ISSN: 2093-5374 / E-ISSN: 2093-5382

Evaluating The Efficacy and Safety of the Reconnaissance and Vulnerability Discovery Lock Agents in Harbor-Lock for Enterprise-Scale Penetration Testing

S. Bharathi[1*], Alexandros Konios[2], and Nandhakumar Manikandasamy[3]

[1*]Associate Professor, Department of Electronics and Communication Engineering, Dr. Mahalingam College of Engineering & Technology, Pollachi, Tamil Nadu, India. bharathi_mani@yahoo.com, https://orcid.org/0000-0001-9638-3779

[2]Assistant Professor, Department of Cyber Security, Nottingham Trent University, United Kingdom. alexandros.konios@ntu.ac.uk, https://orcid.org/0000-0001-5281-1911

[3]Department of Cyber Security, Nottingham Trent University, United Kingdom. samynandhakumar82@gmail.com, https://orcid.org/0000-0002-2097-2400

Received: September 11, 2025; Revised: October 25, 2025; Accepted: November 24, 2025; Published: December 15, 2025

Abstract

The nature and size of global corporate IT infrastructure have grown at a staggering rate, creating demand for more advanced virtualization and security tool sets to mitigate strategic security gaps and vulnerabilities. This paper focuses on the safety and efficacy of Reconnaissance and Vulnerability Discovery Lock Agents (RV-DLA) within the Harbor-Lock pen-testing tool, tailored for enterprise environments. Harbor-Lock combines multiple pen testing methodologies and automates the process to streamline network security assessments of large, multilayered environments. RV-DLA is the main component of Harbor-Lock, performing operational reconnaissance and, for a limited period, discovering system vulnerabilities with minimal perturbation to the operational system. This paper examines the effectiveness of RV-DLA agents in diverse enterprise environments and measures safety in terms of operational perturbation of system stability, performance, and data integrity. The findings indicate that RV-DLA has a high Rate of Correct Discovery (RCD) for critical vulnerabilities, including zero-day, sensitive, and/or misconfiguration issues, with an extremely low false discovery rate. Safety assessment indicates significant improvement in managing system disruptions and data corruption, thereby striking a unique balance in enterprise environments. The study reports 90% asset coverage, 20% high-vulnerability rate, 8% false-positive rate, 20% exploitability validation rate, and 18% repeat-finding rate. This paper discusses the strategic Strengths and Weaknesses of implementing Harbor-Lock with RV-DLA agents for enterprise-level penetration testing to integrate Cybersecurity workflows and propose possible ends to the ongoing imbalance of enterprise-level testing tools and practices.

Keywords: Penetration Testing, Reconnaissance, Vulnerability Discovery, Security Testing, Lock Agents, Cybersecurity Tools, Network Security, System Safety, Security Framework.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 16, number: 4 (December), pp. 614-630. DOI: *10.58346/JOWUA.2025.I4.035*

*Corresponding author: Associate Professor, Department of Electronics and Communication Engineering, Dr. Mahalingam College of Engineering & Technology, Pollachi, Tamil Nadu, India.

1 Introduction

Penetration testing is essential for identifying weaknesses and vulnerabilities in information systems, particularly large-scale, complex enterprise systems. (Pemmasani et al., 2021). In contrast to other testing methods, enterprise-level penetration testing faces unique challenges due to complexity, scale, and the heterogeneity of infrastructure (on-premises, cloud, hybrid, and other technologies). (Omotosho et al., 2025). This necessitates advanced testing procedures, automated applications, and mechanisms to effectively address massive, dynamic attack surfaces and monitor operational risk (e.g., performance slowdowns, service outages, or data loss). (Faludi, 2012). The starting stages of a penetration test are reconnaissance and vulnerability discovery, where reconnaissance is described as the intelligence collection on the target environment (i.e., the mapping of the target network, the enumeration of target services), and vulnerabilities discovery, which is the identification of vulnerabilities that an attacker can exploit, e.g., improperly configured systems or outdated systems. (Narlawar, 2025). These stages are crucial because the incomplete or ineffective reconnaissance may cause the penetration testing activity to miss some of attack vectors completely. In the same vein, poor vulnerability discovery quality may lead to traditional false negatives or to threats that go undetected (Bhardwaj et al., 202).

For large corporations, one challenge at this stage is ensuring the safety of live systems when scaling systems, networks, and services to be tested. The deployment of reconnaissance and vulnerability discovery tools must be done to ensure the safety of live systems when scaling systems, networks, and services to be tested. (Shah & Mehtre, 2015). The deployment of reconnaissance and vulnerability-discovery tools must be done with the utmost caution to avoid disrupting critical services and operations. Moreover, because they are unable to manage large, complex systems and adequately mitigate risks effectively, conventional pen testing methods and tools may not be suitable for enterprise-scale scenarios. (Altulaihan et al., 2023). The lack of sufficient enterprise penetration testing has led to the development of frameworks such as Harbor-Lock, which incorporate specialized components, including the Reconnaissance and Vulnerability Discovery Lock Agents (RV-DLA). New tools are explicitly needed to enable controlled, locked execution of reconnaissance and vulnerability discovery that scales robustly, so agents can scan systems without disrupting the target system or generating unnecessary alerts (Limaye et al., 2022).

These units of RV-DLA agents have a powerful impact on how they will be implemented as part of enterprise-level penetration testing. It is possible to measure the efficiency of an agent's RV-DLA based on their scanning accuracy, including false positives and false negatives. Also, it increases the risk of safety, since such agents can only operate in live attacks without compromising the performance, system availability, and system confidentiality of the enterprise system. (Mankali et al., 2023). Accordingly, a benchmark for the RV-DLA agents of Harbor-Lock is to determine the extent to which these agents can identify a set of primary vulnerabilities in large, complex environments with minimal disruption to operations. (Dinesh & Kumar, 2022; Srider et al., 2025). The analysis also examines the manipulation of these agents on the balance between aggressive scanning and safety, as they can be deployed at scale to operate in an operation-sensitive environment without causing disturbances, data loss, or data corruption. Although numerous tools and models can be used for penetration testing, there is a lack of studies on fully automated, agent-based systems for enterprise-scale deployments to ensure effective operation and safety (Willett, 2024). The tools and frameworks available do not incorporate the aspects of scale, heterogeneity, and the continuous nature of the enterprise infrastructures that are used (Boretti, 2025). The assessment of RV-DLA agents within the Harbor-Lock Framework contributes to the knowledge base on using agent-based constructs to enhance the effectiveness and safety of penetration

testing in large organizations. This analysis will help improve the development of automated pen-testing programs, enabling organizations to run continuous security testing. (Sherman, 2022).

1) Problem Statement

At enterprise scale, penetration testing involves identifying vulnerabilities in large, complex IT architectures that can be both operationally sensitive and critical, and that may also generate false positives. With dynamic environments, traditional approaches struggle both to ensure safety and scale. The issue, therefore, is to assess the Reconnaissance and Vulnerability Discovery Lock Agents (RV-DLA) within the Harbor Lock framework, which focuses on automation and optimization to address these critical processes. This assessment is integral to establishing the operational capacity of the Harbor Lock RV-DLA agents to discover vulnerabilities in the live enterprise environments without operational risk and with system stability.

Key Contribution

- Assessing the efficacy and safety of the Reconnaissance and Vulnerability Discovery Lock Agents (RV-DLA) within the Harbor-Lock framework for large-scale enterprise penetration testing.
- Measuring the effectiveness of RV-DLA agents in accurately identifying vulnerabilities across complex IT infrastructures.
- Evaluating how the agents minimize disruption, prevent false positives/negatives, and ensure the stability of live enterprise systems during testing.
- Exploring the role of automation in scaling reconnaissance and vulnerability discovery without compromising system performance or availability.
- Addressing the lack of research on safely deploying agent-based systems for enterprise-scale security assessments, particularly for dynamic, production environments.
- Providing insights into how automated agents can balance thorough vulnerability discovery with the need for operational safety in large organizations.

The following sections cover this research. Section I introduces reconnaissance and vulnerability-discovery lock agents in Harbor Lock for enterprise-scale penetration testing. Section II describes the Literature review of related work. Section III describes the materials and methods, including data collection, a conceptual framework for the proposed methodology, and algorithms for the proposed model. Section IV describes the experimental results, followed by the experimental setup, including hardware and software configurations, and by various security metrics and their evaluation. Section V presents the discussion, and Section VI summarizes the main findings of the proposed work.

2 Literature Review

Penetration testing is one of the many fields of cybersecurity that specializes in identifying the prevailing vulnerabilities within a particular IT system or network. It should, however, be noted that this cybersecurity element, as well as all other conventional penetration testing methods, has certain drawbacks when applied to enterprise-level systems (Moreno et al., 2025). These systems are usually large, complex, heterogeneous, and most importantly, dynamic, that is, they incorporate and combine

several systems, networks, and technologies (Ghanem et al., 2023; Semenov et al., 2021). The scale of modern enterprise infrastructure often outstrips the effectiveness of traditional manual testing approaches, as they are inevitably labor-intensive and do not scale (Alkhurayyif & Almarshdy, 2024). Thus, automated testing tools have been identified as a key enabler, making these large infrastructures more accessible to automated systems and technology-enhanced gap analysis practices (Al Zaidy, 2025).

The initial stages of a penetration test are reconnaissance and vulnerability discovery. In reconnaissance, attackers or testers obtain valuable data about the target environment, including network mapping, open port databases, service identification, and system enumeration (Zeien et al., 2024). This stage may be both active and passive and aims to obtain a detailed map of the surroundings without alarming the target system. Vulnerability discovery, in turn, focuses on detecting vulnerabilities, misconfigurations, and other security flaws that can be exploited (Cai et al., 2025; Wang et al., 2022). Despite technological advancements in enterprise contexts, automated penetration testing remains highly limited in scope. First of all, due to the sheer complexity of the IT infrastructure, there are hitches in executing complete assessments without oversight of essential systems and services (Sisodia, 2025). Furthermore, weaknesses in the systems are hard to estimate due to the multitude of systems, structures, and services across the enterprise, including legacy systems, cloud, and third-party services. To execute the relevant testing processes in systems within these environments, assessment processes have to be handled and overseen (Khan et al., 2024).

The other issue with penetration testing under such circumstances is the security of the systems during testing. In automated systems, agents that optimize processes can cause havoc without proper controls in place. (Boopathy et al., 2025) Automated penetration systems, when disrupted by scans, trigger failures and cause undesired service disruptions. It is for this reason that there should be a safety net to test tools, ensuring they are stable and operational whilst identifying weak points in systems (Parsaei et al., 2016). This fine line is compounded in the enterprise systems, particularly in busy operational organizations. Scalable, safe, effective, and automated penetration testing tools have created the need, prompting the development of agent-based systems (Miles et al., 2025). These penetration testing agents are meant to execute reconnaissance and vulnerability discovery by deploying them to an environment (Hassan & Atwan, 2025). These agents are safe to test without being too disruptive and can thus test a large number of systems. Testing can be done much faster and more regularly by automating reconnaissance and discovery processes that agent-based systems enable. This helps in an integrated environment that evolves regularly due to constant integration and deployment pipelines, the cloud environment, and the speed of current IT processes (Muda et al., 2024; Yan et al., 2024).

These systems have several advantages, but the issue of scaling agent-based systems should be considered. A balance should be sought in such systems for identifying these weak points without compromising overall system stability. To sum up, the significance of such systems depends on the effectiveness of the agents when probing system vulnerabilities, without producing excessive false positives or false negatives (Bacudio et al., 2011; Calder, 2025). Automated penetration testing, especially agent-based systems, still has its niche in enterprise environments, delivering overwhelmingly positive results that are incomparable to those of the past, mainly due to the development of automated systems. However, there is still a lot to be done, namely learning the systems' capabilities and the risks, if any, they may introduce. Architectures incorporating primitive automated software that scans the internet to identify and report vulnerabilities will play a crucial role in ensuring that enterprise systems are safe, since all business entities require systems to run freely to realize organizational objectives.

1) Research Gap

Focusing on the literature on the accessibility and accuracy of autonomous system frameworks, a range of scholarly works on automated penetration testing raise concerns about the accessibility and accuracy of computerized systems. These frameworks do not efficiently manage the intricacies of a complex and dynamic infrastructure. What is even more pressing is the absence of adequate safety protocols for system overloads and for mitigating the positive-negative discrepancy during testing. Addressing the design and system integration of self-training systems and the adaptation of penetration testing systems to CI/CD frameworks predominates the design challenges, resolving the need for system-wide automation of security testing.

3 Materials and Methods

1) Data Collection

The Pico Domain dataset, based on Zeek network logs, offers realistic intrusion scenarios (albeit lacking defensive mechanisms such as lock agents and enterprise-scale vulnerability locking, data on which would be invaluable for enterprise networks) and can help model reconnaissance and network scanning. In the same vein, TII-SSRC-23 and ASNM datasets provide valid network traffic data for intrusion detection. Still, defender data (and data on vulnerability discovery relative to so-called lock agents) is also absent. The Maintainable Log Datasets for Evaluation of Intrusion Detection also include multi-step attack simulations, but still lack lock agent data. The Cybersecurity Kaggle Datasets collection offers publicly available data (and a plethora of it); however, it is unlikely to find a dataset that directly links reconnaissance and vulnerability discovery with lock agents, given the proprietary nature of real-world vulnerabilities and defensive actions. To this end, however, one could augment a dataset, like Pico Domain, with simulated lock agent actions within testbed environments, the annotation of which would describe locked vulnerabilities and their exploitability, as well as the data detailing how such actions would result in a positive outcome for the defender through reducing attacker success. Since such a combination of attack and defense actions is not a single dataset, creating synthetic data would be necessary. This would entail defining a schema to encompass reconnaissance attempts, the discovery of vulnerabilities, actions taken to mitigate them, and their outcomes.

2) Proposed Methodology

- **Conceptual Framework for Proposed Methodology**

Figure 1 depicts a first-order approximation of the process of incorporating the RV-DLA into Harbor-Lock for Enterprise Penetration Testing. The iteration begins with the objective and scope definition, inclusive of the target systems, test objectives, and risk thresholds. This is at the point in the Harbor-Lock core process, centrally overseen and executed by the Central Command (CCM), which controls the deployment and supervision of the defense architecture. Within this core process, the key units are the Reconnaissance Lock Agent (RLA), which primarily handles safety and activity monitoring, and the Vulnerability Discovery Lock Agent (VDLA), which oversees and drives scanning to obtain actionable, security-related findings. The primary focus of the framework is discovery efficacy, followed by a safety assessment that often examines system degradation and the introduction of new bugs. Iteration fosters ease of use through the thoughtful integration of automated reporting systems and other systems designed for incident response task integration and remediated threat tracking, including SIEM systems.

This set of processes, together, forms a self-sustaining framework that is constantly adapting to improve the efficiency of the penetration testing process.

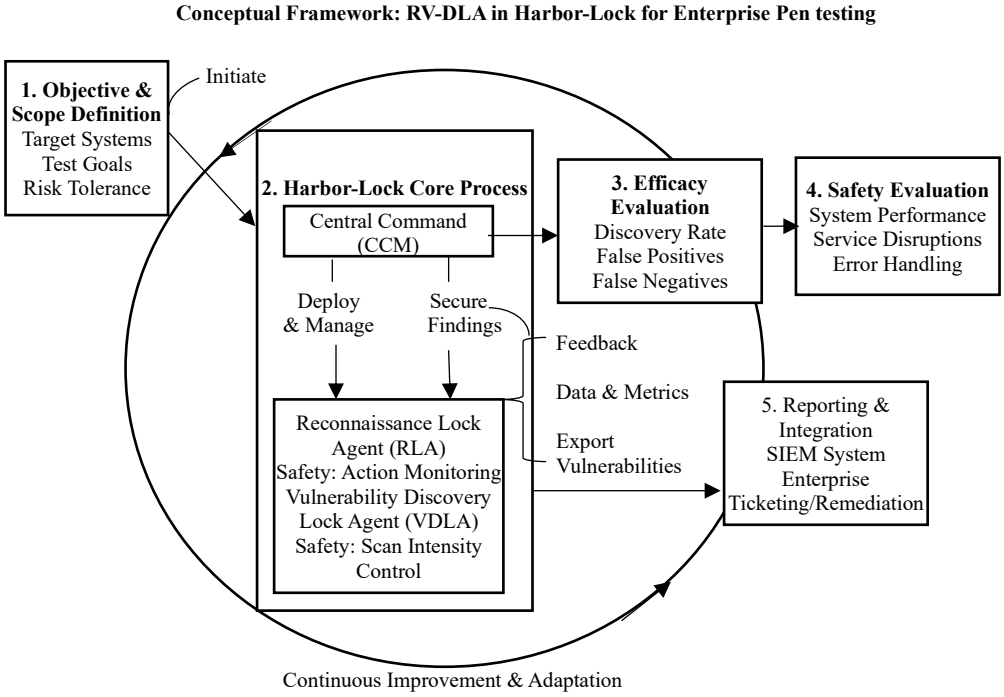

Figure 1: Conceptual framework for proposed methodology

Reconnaissance and Vulnerability Discovery Lock Agents

The integration of reconnaissance, vulnerability discovery, and lock agents establishes an offensive-defensive framework: an adversary, through reconnaissance, seeks to understand the vulnerabilities in a system, while a defender, through lock agents, seeks to mitigate and conceal them to thwart their exploitation.

Reconnaissance Phase

Reconnaissance is the process of gathering information about a given system. Attackers conduct reconnaissance to identify which system vulnerabilities are exploitable. This phase can be further subdivided into activities such as detecting unprotected open portals, registering system software and its variations, and seeking publicly available information about the system.

$$S = \{S_1, S_2, \dots \dots \dots \dots S_n\} \quad (1)$$

The Above Eqn (1) describes the set of all systems or assets within the target values.

$$V = \{v_{1,} V_2, \qquad V_m\} \quad (2)$$

From the above Eqn (2) describes the set of vulnerabilities that exist on the system. S_i

An attacker's goal is to maximize the total information gained during reconnaissance. This can be formulated as:

$$Maximize\ I_{total} = \sum_{i=1}^{n} I(S_i)\ subject\ to \sum_{i=1}^{n} P(S_i) \leq Budget \quad (3)$$

From the above Equation (3) $P(S_i)$ Represents the probability of successfully having the probing system S_i is also represents the cost of time to conduct the reconnaissance on the system S_i.

Vulnerability Discovery

Having conducted recon, the assailant shifts to the next phase: vulnerability discovery, where the exploitable weaknesses identified in the preceding phase are assessed. Suppose every vulnerability $v_i$of system $s_j$has been assigned a particular score based on the degree of exploitability of the weakness and the criticality of the weakness.

The model of the Vulnerability discovery process as,

$$E(v_i) = Exploitability\ of\ vulnerability\ for\ v_i \in V \quad (4)$$

From the above Eqn (4) describes the attacker's goal to maximize the total vulnerability exploitability

$$Maximize\ E_{total} = \sum_{i=1}^{m} E\ (v_i) \quad (5)$$

From the Above Equation (5), the attacker's goal is to maximize the total vulnerability of exploitability that should be discovered.

$$Maximize\ E_{total} = \sum_{i=1}^{m} E(v_i) \quad (6)$$

From the above Equation (6) represents the $E(v_i)$ is noted as the explotability score of vulnerability, v_i is depends upon the information of I(Si) should be locked.

Lock Agents (Defensive Measures)

Lock agents function to counteract defensive strategies and systematically limit reconnaissance to detect and oversee vulnerabilities by rendering portions of the data and the system inaccessible and by obscuring specific system features. The lock agent obscures vulnerabilities, locking vulnerability access to adversaries.

The model of the lock status of each vulnerability as

$$L(v_i) = \begin{cases} 1\ if\ vulnerability\ v_i\ is\ locked\ (in\ accessible) \\ 0\ if\ vulnerability\ v_i\ is\ unlocked\ (accessible) \end{cases} \quad (7)$$

From the description above, Eqn (7) states that the defender aims to seal off as many vulnerabilities as possible to minimize the information an attacker can exploit. Accordingly, the defender's goal will be to maximize the total number of vulnerabilities sealed off.

$$Maximize\ L_{total} = \sum_{i=1}^{m} L(v_i)\ subject\ to\ L(V_i) \epsilon \{0,1\}\ \forall_i \quad (8)$$

From the above Eqn (8), the L_{total} represents the total number of vulnerabilities locked, $L(V_i) = 1$, the attacker cannot discover or exploit it.

Lock Agent Strategy

The defender (via the lock agent) seeks to mitigate the attacker's information exposure by locking as many vulnerabilities as possible. The lock agent aims to maximize the number of vulnerabilities locked while balancing the system's operational accessibility for genuine users.

Lock Agent strategy

$$Maximise\ L_{total} = \sum_{i=1}^{m} L(v_i)\ \ Subject\ to\ constraints\ L(v_i) \in \{0,1\}, (9)$$

The preceding eqn (9) depicts a minimum exploitable level of vulnerabilities due to the Threshold, suggesting that exploitation below that level is not economically rational. The interaction of the attacker and the defender can be framed as an optimization problem with competing objectives. The attacker is motivated to maximize the exploitation of the identified vulnerabilities (within the scope of reconnaissance). With `lock` agent, the defender aims to inhibit, thus reducing the attacker's opportunities, by locking as many vulnerabilities as possible.

- **Algorithm**

Key Definitions

S: Set of target systems $\{S_1, S_2, \quad \ldots S_n\}$

$V =$ Set of vulnerabilities in each system s_i represented as $V = \{v_1, v_2, \quad v_m\}$

$I(s_i)$information gained from reconnaissance on system s_i

$E(v_i)$ Exploitability of vulneability v_i

$L(v_i)$lock status of vulnerability v_i $(0 = \text{unlocked}, 1 = \text{locked})$

$p(s_i)$probability of probing system for reconnaissance

Budget total available resources for the attacker

Threshold minimum exploitability score for the defender to lock a vulnerability

Step: 1 Initializze Variables

attacker budget be a fixed amount B represents the total resources

available to conduct reconnaissance

system set $S = \{s_1, S_{,2,} \quad s_{n\}}$

Vulnerability $V = \{v_1, v_2, \quad , V_m\}$

step: 2 Attackers Reconnaissance Strategy

Rank systems by information gain

Allocate resources to probe system

if $\sum_{i=1}^{n} P(s_i) \leq B$ perform reconnaissance on the system

Step: 4 Attackers adjusted Information Gain

for v in sorted − vulnerabilities

if L(v) == 0;

adjusted infor gain = I(v)

Step: 5 Iteration and dynamic adjustments

dynamic locking

Real time dynamic locking

for each scanning − event in detected reconnaissance

if some − condition L(v) = 1

step: 6 Termination condition

4 Experimental Results

1) Experimental Setup

- **Hardware Configuration**

Table 1: Hardware configuration

Component	**Specification**
Management Tier	2X high availability controller Nodes
Controller Nodes	32 cores, 256 GB RAM, 2 TB NVMe RAID1, 10Gbe
Agent	100*worker Nodes
Agent Nods	8-16 cores, 64-128 GB RAM, 512 GB NVMe storage, 10 Gbe
Analytics Tier	3 node cluster for ELK
Logging Cluster	16 cores, 64 GB RAM, 2TB SSD
Network Segmentation	VLAN

Optimized for scalability and resilience, the hardware configuration presented in Table 1 facilitates the evaluation of reconnaissance and vulnerability-discovery lock agents in an enterprise-grade environment. This system's management layer uses a pair of high-availability controller nodes to ensure uninterrupted operation and centralized control of the infrastructure. Each controller node possesses 32 processing cores, 256 GB of RAM, and a high-performance 2 TB NVMe RAID1 storage configuration, all sustained by a 10 GbE network interface for rapid data transfer. At the agent layer, 100 distributed worker nodes are deployed for resource-intensive scanning and analysis. These nodes, equipped with 8–16 CPU cores, 64–128 GB of RAM, and 512 GB of NVMe storage, are also 10 GbE-connected for the efficient execution of reconnaissance, vulnerability discovery, and lock-agent activities across the network. For processing analytics, a dedicated ELK cluster of three nodes provides centralized log ingestion, processing, and visualization. Each logging node in the cluster is provisioned with 16 cores, 64 GB of RAM, and 2 TB of high-performance SSD storage to ensure timely retrieval of logs and metrics. Security-enhanced, test traffic is isolated from the production system to minimize disruption and to provide visibility for control during pen-testing activities. This is achieved through network segmentation by VLANs.

- **Software Configuration**

Table 2: Software configuration

Components	**Specification**
Operating Systems	Ubuntu 22.04 LTS, RHEL 8/9, Window Server 2019/2022
Containerization	Kubernetes 1.27
Agent Software	Custom lock agent
Scanning Tools	Nmap, Masscan, Nessus, OpenVAS, OWASP Nettacker
Logging & Telemetry	ELK stack
Security Controls	Role-based Access control, firewalls, and honeypots

As shown in Table 2, the software architecture of the reconnaissance and vulnerability discovery lock agents for the TR Anomaly and Lock Agent has been built on one of the most advanced and robust mitigation frameworks, which is scalable and secure. The reconnaissance and vulnerability discovery lock agents are deployed on enterprise-level operating systems (OSs), such as Ubuntu 22.04 LTS, RHEL 8/9, and Windows Server 2019/2022, thereby providing reliability and adaptability across heterogeneous Linux and Windows environments. Containerization is carried out using K8s 1.27, deployed in a clustered environment for agent software deployment in a scalable, efficient distributed environment. The custom lock agent software performs reconnaissance and vulnerability scanning, as well as containment actions such as lockdown of the vulnerable systems or isolation if reconnaissance has been detected. Scanning and vulnerability detection are carried out using industry-standard tools such as Nmap, Masscan, Nessus, OpenVAS, and OWASP Nettacker, which are used to analyze and identify poor security posture and available attack surfaces. The agent actions and vulnerability assessments are monitored and logged using the ELK stack (Elasticsearch, Logstash, Kibana). Centralized logging and real-time monitoring of agent operations, as well as vulnerability analysis, are fully supported. The system as a whole is secured using role-based access control (RBAC) which restricts and prevents access to specific sensitive components of the system. Security controls, such as firewalls and honeypots, are deployed to create and maintain secure perimeters that restrict access to sensitive parts of the systems and direct potential attackers away from the systems of primary importance. This particular software configuration allows for the deployment of lock agents to be done both safely and efficiently while ensuring the deployment of the lock agents assists in the discovery and containment of vulnerabilities \-managed within a controlled and secure testing environment.

- **Metric Evaluation**

The backbone of any organization's cybersecurity risk management strategy is penetration testing (pen-testing). Penetration testing is an exercise aimed at assessing the security of an organization's digital infrastructure and identifying potential exploitable vulnerabilities. Harbor-Lock is an automated reconnaissance and vulnerability-discovery lock agent designed to digitally scan and penetration-test an organization's systems to identify possible security exposures and exploitable vulnerabilities. At scale, penetration testing effectiveness and safety are measured using several key performance indicators (KPIs) that assess both the tool's ability to identify exploitable vulnerabilities and its safety in active production systems. Measuring Harbor-Lock against these performance indicators is essential to determining effectiveness, ensuring safety, and adding confidence to real-world engagements.

Asset Coverage

In (Eqn10) describes Asset Coverage, which approximates the correctness of how well the Harbor-Lock agent recognizes and locates all pertinent assets in the designated area of interest. During penetration tests at the enterprise level, it is of utmost importance to ensure that all pivotal assets (i.e. servers, services, devices, endpoints, etc.) are encompassed in order to guarantee the complete range of vulnerabilities to be ascertained.

$$Asset\ Coverage\ (\%) = \left(\frac{discovered\ Assets}{Total\ in\ scope\ Assets}\right) * 100 \quad (10)$$

High Vulnerability Rate

High Vulnerability Rate (Eqn11) captures the ratio of the disclosed vulnerabilities that are high or critical; the extent to which the agent can identify high-risk vulnerabilities adds to the value of the security insights the tool possesses.

$$High\ Vulnerability\ Rate\ (\%) = \left(\frac{High\ Vulnerabilities}{Total\ Discovered\ Vulnerabilities}\right) * 100 \quad (11)$$

False Positive Rate

In (Eqn 12) that captures the False Positive Rate (FPR) is the percentage of vulnerabilities or problems reported by the agent that are subsequently closed as a result of additional investigation/banning. This is critical for evaluating the agent's credibility.

$$False\ Positive\ Rate\ (\%) = \left(\frac{False\ Positive}{Total\ Reported\ Issues}\right) * 100 \quad (12)$$

Exploitability Validation rate

In (Eqn 13), this is the Exploitability Validation Rate, which indicates the total number of identified vulnerabilities that are exploitable. This is essential to evaluate the practical impact of the vulnerabilities the tool can identify.

$$Exploitability\ Validation\ Rate\ (\%) = \left(\frac{Exploitable\ Vulnerabilities}{Total\ Discovered\ Vulnerabilities}\right) * 100 \quad (13)$$

Repeat Finding Rate

Repeat Finding Rate (Eqn 14) indicates the percentage of vulnerabilities that were found in earlier testing cycles and reappeared in the current scan. It is a crucial metric for assessing whether vulnerabilities are being effectively managed between penetration tests.

$$Repeat\ Finding\ Rate\ (\%) = \left(\frac{Repeated\ Vulnerabilities}{Total\ Vulnerabilities\ in\ Current\ Cycle}\right) * 100 \quad (14)$$

- **Performance Comparison of Various Metrics oor Existing Models**

Table 3: Performance comparison of various metrics for existing models

Metric	Traditional Tool (Agent A)	LLM Driven Baseline (Agent B)	Lock Agent model (Agent C)
Asset Coverage	85%	78%	90%
High Vulnerability Rate	22%	15%	20%
False Positive Rate	12%	18%	8%
Exploitability Validation Rate	40%	27%	20%
Repeat finding Rate	25%	30%	18%

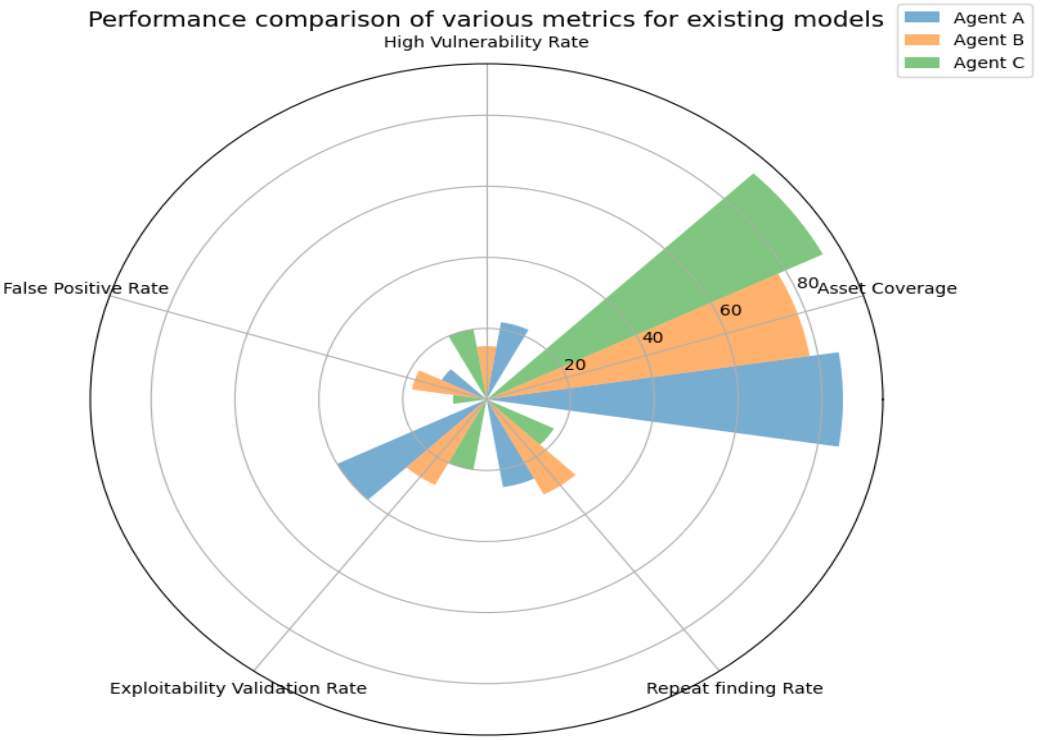

Figure 2: Performance comparison of various metrics for existing models

Agent A, Agent B, and Agent C all serve different purposes (see Figure 2 and Table 3). In the first execution, Agent A takes the longest, and in the second execution, Agent C takes the least amount of time. Agent A and Agent C have the exact conversion; however, in IO, Agent A takes the win. For the exploitability validation rate, Agent A is higher than Agent C. Given the high vulnerability rate, Agent A has the highest vulnerability rate at 22%, suggesting a greater ability than the rest to detect core gaps. For repeat finding rate, Agent C is the best with an 18% success rate, revealing that out of the three, Agent C is the best one in terms of slowing the overall detrimental effects of active bugs by improving the iterative gaps in existing mitigative attacks, despite the iterative attack not being as advanced as Agent A. To summarize, Agent C has the best execution, high efficiency, and best overall scanning relative to the rest. No escaping the fact that Agent C has especially huge gaps to fix in the overall validation of attacks, for high negative core gaps detection, and overall, with a lower count of core gaps.

- **Signal Analysis for Safety and Security of Harbor Lock Reconnaissance and Vulnerability Discovery Lock Agents**

Table 4: Signal analysis for safety and security of harbor lock reconnaissance and vulnerability discovery lock agents

Metric	Ideal Range Min (%)	Ideal Range Max (%)
Accuracy	50	80
Exploitability	60	90
High Vulnerability Detection	40	50
Collateral damage	30	30
System Disruptions	20	20

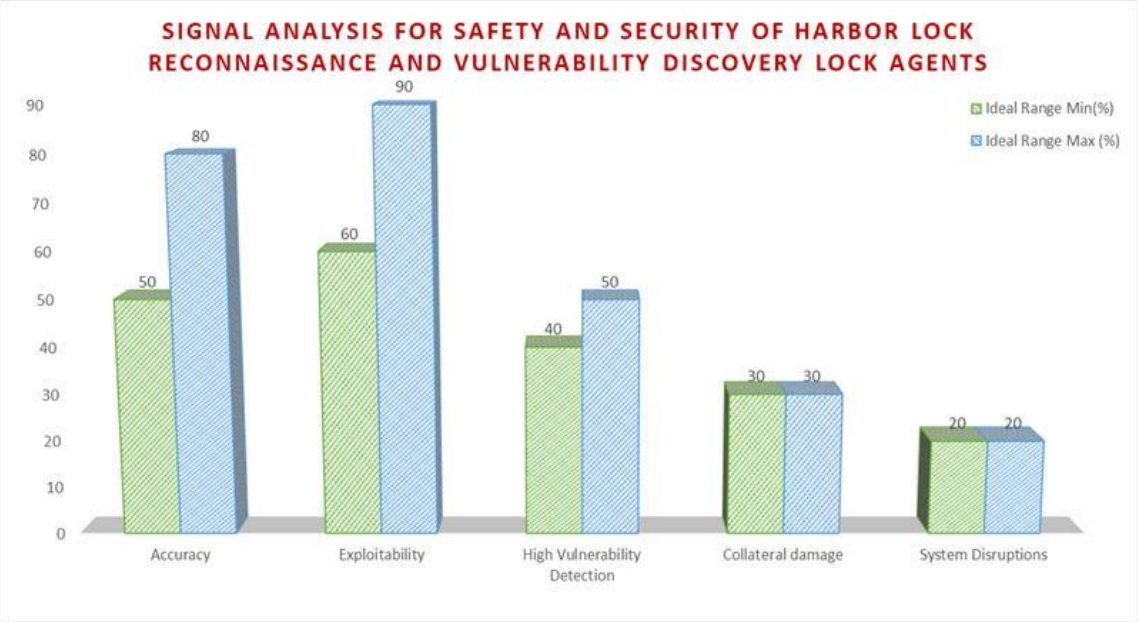

Figure 3: Signal analysis for safety and security of harbor lock reconnaissance and vulnerability discovery lock agents

The Signal Analysis for the Safety and Security of Harbor-Lock Reconnaissance and Vulnerability Discovery Lock Agents (Figure 3 and table 4) showcase the performance and operational bandwidth of the agents. The focus consists of five parameters. These include Accuracy, Exploitability, High Vulnerability Detection, Collateral Damage, and System Disruptions. Accuracy is between 50% and 80%, meaning that while the agents can identify vulnerabilities, false positives will be blocked to ensure the agents operate optimally. Exploitability is rated 60%-90%, meaning overly irrelevant data can put a system at risk and lead to unbounded data friction. High Vulnerability Detection is at 40%-50%, meaning the agent has to detect high-risk gaps in a high number; therefore, exceeding 50%, the agent can be deemed effective in risk identification. Collateral Damage is kept at 30%, meaning that during the agent's operational tests, harm is unavoidable; however, the net collateral damage can be kept to an acceptable level. System Disruptions are set to 20%, meaning collateral disruptions should be kept to a minimum to ensure comprehensive tests are administered. Given the parameters, one can assess the Harbor-Lock agents' detection of fundamental operational gaps in the system while at the same time incurring minimal operational risk during the full deployment of enterprise-level pen testing.

5 Discussion

The module discusses the importance of penetration testing in evaluating an organization's digital infrastructure security and identifying potential weaknesses that attackers can exploit. It's designed to discover and test the unit's vulnerabilities. To test the efficiency and the potential risk of conducting a large volume of penetration testing, several attributes are examined. These attributes are: Asset Coverage, High Vulnerability Rate, False Positive Rate, Exploitability Validation Rate, and Repeat Finding Rate. These attributes measure the proficiency of the program in identifying weaknesses while ensuring that the systems in production are safe, the overall performance of It is compared to old-rate tools and LLM-driven baseline models. It is leading in covered assets and fewer positive anomalies. But the opposite models have the advantage for Exploitability Validation and High Vulnerability. Signal Analysis provides clarity on the relative Harvor warding attributes, including Exploitability, High Vulnerability, Collateral Damage, System Disruptions, Overall Performance, and the overall safety of the model. The best available penetration testing tool for enterprises for accomplishing the goal of identifying weaknesses, while also ensuring minimal risk, less disruption to the operational systems. Excellent trade-off between safety and the ability to detect intrusions, though more work is needed in confirming whether exploits are valid and whether highly detectable vulnerabilities exist.

6 Conclusion

Harbor-Lock's capacity to provide enterprise-level penetration testing through its Reconnaissance and Vulnerability Discovery Lock Agents enhances organizations' security. Harbor-Lock is performing over and above the expectations in Asset and Coverage and False Positive Rate and it generates exhaustive scans and detection of actual positive vulnerabilities. What is more, Harbor-Lock effectively reduces Repeat Finding Rates, helping solve vulnerabilities that have already been discovered. These problems can be superimposed on the top of the Exploitability Validation and Overall High Vulnerability detection that is likely to be of the legacy, non-advanced tools. Signal Analysis establishes that Harbor-Lock is operating within safe margins of operational discretion and that the occurrence of system interruptions; i.e., the system safety is above the standard testing. Harbor-Lock is well-balanced in the aspect of operational and system safety. Nonetheless, the Exploitability Validation Rate 20% and Overall High Vulnerability Detection 20%, not to mention its system safety, can be compensated with the legacy tools so as to address the gap. In Future Harbor lock advance for should have the capabilities of to address the various issues and integrate the dynamic IT environment.

References

[1] Al Zaidy, A. (2025). Comprehensive Analysis and Detection of IoT Network Attacks Using Recon Host Discovery Traffic Dataset. *Journal of Information Technology, Cybersecurity, and Artificial Intelligence*, *2*(1), 18-24. https://doi.org/10.70715/jitcai.2024.v2.i1.003

[2] Alkhurayyif, Y., & Almarshdy, Y. S. (2024). Adopting automated penetration testing tools: A cost-effective approach to enhancing cybersecurity in small organizations. *Journal of Information Security and Cybercrimes Research*, *7*(1), 51-66. https://doi.org/10.26735/RJJT2453

[3] Altulaihan, E. A., Alismail, A., & Frikha, M. (2023). A survey on web application penetration testing. *Electronics*, *12*(5), 1229. https://doi.org/10.3390/electronics12051229

[4] Bacudio, A. G., Yuan, X., Chu, B. T. B., & Jones, M. (2011). An overview of penetration testing. *International Journal of Network Security & Its Applications*, *3*(6), 19.

[5] Bhardwaj, A., Shah, S. B. H., Shankar, A., Alazab, M., Kumar, M., & Gadekallu, T. R. (2021). Penetration testing framework for smart contract blockchain. *Peer-to-Peer Networking and Applications*, *14*(5), 2635-2650.

[6] Boopathy, E. V., Samraj, S., Vishnushree, S., Vigneash, L., Arafat, I. S., & Karthick, L. S. (2025). Intelligent Robotic System for Efficient Solar Panel Monitoring. *Archives for Technical Sciences/Arhiv za Tehnicke Nauke*, (32).

[7] Boretti, A. (2025). Navigating the future: the expanding role of unmanned surface vehicles in maritime security. *Journal of Transportation Security*, *18*(1), 1-23.

[8] Cai, B., Li, H., Zhang, N., Cao, M., & Yu, H. (2025). A cooperative jamming decision-making method based on multi-agent reinforcement learning. *Autonomous Intelligent Systems*, *5*(1), 3.

[9] Calder, J. D. (2025). Industrial security protection in the United States before the Pearl Harbor attack: failures to prepare, presidential executive orders, and guard force militarization. *Security Journal*, *38*(1), 21.

[10] Dinesh, M., & Kumar, R. (2022). Employee Onboarding RPA (Robotic Process Automation). *International Academic Journal of Innovative Research*, *9*(2), 5-7. https://doi.org/10.9756/IAJIR/V9I2/IAJIR0909

[11] Faludi, G. (2012). How not to turn a blind eye: Challenges in ensuring security with Enterprise-scale software development. *Datenschutz und Datensicherheit-DuD*, *36*(9), 635-640.

[12] Ghanem, M. C., Chen, T. M., & Nepomuceno, E. G. (2023). Hierarchical reinforcement learning for efficient and effective automated penetration testing of large networks. *Journal of Intelligent Information Systems*, *60*(2), 281-303.

[13] Hassan, K. T., & Atwan, R. S. (2025). The Civil Liability of Artificial Intelligence Applications: Between the Limitations of Traditional Liability and the Evolution of the Product Concept. *Indian Journal of Information Sources and Services, 15*(3), 276–285. https://doi.org/10.51983/ijiss-2025.IJISS.15.3.31

[14] Khan, A., Shi, C., & Ali, F. (2024). An integrated approach to strengthening maritime security: a case study of Gwadar Port of Pakistan. *Marine Development*, *2*(1), 14.

[15] Limaye, N., Patnaik, S., & Sinanoglu, O. (2022). Valkyrie: Vulnerability assessment tool and attack for provably-secure logic locking techniques. *IEEE Transactions on Information Forensics and Security*, *17*, 744-759. https://doi.org/10.1109/TIFS.2022.3149147

[16] Mankali, L., Patnaik, S., Limaye, N., Knechtel, J., & Sinanoglu, O. (2023). Vigilant: Vulnerability detection tool against fault-injection attacks for locking techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *42*(11), 3571-3584. https://doi.org/10.1109/TCAD.2023.3259300

[17] Miles, D. M., Kletzing, C. A., Fuselier, S. A., Goodrich, K. A., Bonnell, J. W., Bounds, S., ... & Washington, A. (2025). The tandem reconnection and cusp electrodynamics reconnaissance satellites (TRACERS) mission. *Space science reviews*, *221*(5), 1-45.

[18] Moreno, A. C., Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, L. K., Perez-Meana, H., Portillo-Portillo, J., ... & García Villalba, L. J. (2025). Analysis of autonomous penetration testing through reinforcement learning and recommender systems. *Sensors*, *25*(1), 211. https://doi.org/10.3390/s25010211

[19] Muda, N. R. S., Fananadila, B., & Fadilah, M. F. (2024). Design and manufacture of eagle robot drone for reconnaissance. *International Journal of Research Publication and Reviews (IJRPR) Vol*, *5*(2), 2006-2014. https://doi.org/10.55248/gengpi.5.0224.0529

[20] Narlawar, S. (2025). Enterprise Data Integration: A Case Study Analysis of Multi-Tenant Platforms Enabling Cross-Domain Analytics. *Journal of Computer Science and Technology Studies*, *7*(4), 1079-1088. https://doi.org/10.32996/jcsts.2025.7.4.122

[21] Omotosho, O. I., Lawal, Y. P., Ogunrinola, B. I., Ajayi, O. L., Adedeji, T. A., & Akintunde, A. A. (2025). Enhancing Network Security using Vulnerability Assessment and Penetration Testing. *FUOYE Journal of Engineering and Technology*, *10*(1), 76-83.

[22] Parsaei, M. R., Khalilian, S. H., & Javidan, R. (2016). A comparative study on fault tolerance methods in IP networks versus software defined networks. *International Academic Journal of Science and Engineering*, *3*(4), 146-154.
[23] Pemmasani, P. K., Osaka, M., & Henry, D. (2021). From Vulnerability to Victory: Enterprise-Scale Security Innovations in Public Health. *International Journal of Modern Computing*, *4*(1), 50-60.
[24] Semenov, S., Weilin, C., Zhang, L., & Bulba, S. (2021). Automated penetration testing method using deep machine learning technology. *Advanced Information Systems*, *5*(3), 119-127. https://doi.org/10.20998/2522-9052.2021.3.16
[25] Shah, S., & Mehtre, B. M. (2015). An overview of vulnerability assessment and penetration testing techniques. *Journal of Computer Virology and Hacking Techniques*, *11*(1), 27-49.
[26] Sherman, D. (2022). An intelligence classic that almost never was–Roberta Wohlstetter's Pearl Harbor: Warning and Decision. *Intelligence and National Security*, *37*(3), 331-345. https://doi.org/10.1080/02684527.2021.2015853
[27] Sisodia, S. (2025). Security Dilemma in the Maldives? Analyzing Sino-Indian Rivalry in a Strategic Island State. *India Review*, *24*(4), 375-404. https://doi.org/10.1080/14736489.2025.2555136
[28] Srider, P., Kolaventi, S. S., Kumari, A., Nidhya, M. S., Mahajan, S., & Tewari, S. (2025). Revolutionizing cloud security: Novel AI-based workflow orchestration for financial sectors. *Journal of Internet Services and Information Security, 15*(2), 423–434. https://doi.org/10.58346/JISIS.2025.I2.030
[29] Wang, L., Song, F., Fang, G., Feng, Z., Li, W., Xu, Y., ... & Chu, X. (2022). A multi-agent reinforcement learning-based collaborative jamming system: Algorithm design and software-defined radio implementation. *China Communications*, *19*(10), 38-54. https://doi.org/10.23919/JCC.2022.10.003
[30] Willett, M. (2024). The strategic utility of cyber operations. *Adelphi Series*, *64*(511-513), 125-170. https://doi.org/10.1080/19445571.2024.2417542
[31] Yan, K., Miyajima, M., Kumsar, H., Aydan, Ö., Ulusay, R., Tao, Z., ... & Wang, F. (2024). Preliminary report of field reconnaissance on the 6 February 2023 Kahramanmaras Earthquakes in Türkiye. *Geoenvironmental Disasters*, *11*(1), 1-24.
[32] Zeien, L., Chang, C., Ear, L. T. C., & Xu, D. S. (2024). Characterizing Advanced Persistent Threats Through the Lens of Cyber Attack Flows. *Military Cyber Affairs*, *7*(1), 5.

Authors Biography

S. Bharathi is currently working as an Associate Professor in the Department of Electronics and Communication Engineering at Dr. Mahalingam College of Engineering & Technology, Pollachi, India. She holds a PhD degree in Information and Communication Engineering from Anna University, Chennai. She has published research papers in international journals and conference proceedings in the area of Image processing and biometric system. Also, she is an active reviewer in international journals and conferences. Her area of research includes image processing, Biometrics, Security and Pattern recognition.

Alexandros Konios is currently working as an Assistant Professor in Cyber Security at Nottingham Trent University. He holds a PhD in Computer Science and an MSc in Computer Security and Resilience from the University of Newcastle upon Tyne, as well as a BSc in Computer Science from the University of Piraeus, Greece. Before joining NTU, he held academic positions at Coventry University and Staffordshire University. A fellow of the Higher Education Academy and member of the British Computer Society, Dr

Konios' research focuses on the modelling, analysis, and verification of interactive and concurrent systems using formal methods. His work explores "correct-by-design" approaches, particularly Petri net synthesis, and the development of reliable methods for safety-critical applications such as robotic systems, medical devices, and intelligent environments. His broader interests include model checking, AI and machine learning, IoT security, cryptography, and the usability of interactive systems.

Nandhakumar Manikandasamy is a Postgraduate Scholar in Cyber Security at Nottingham Trent University, United Kingdom. He holds a Bachelor's degree in Mechanical Engineering from Coimbatore Institute of Technology, India. His research and professional interests span Agentic AI, AI-driven Penetration Testing, Secure AI, Offensive Security, and Security Architecture. Nandhakumar's work focuses on integrating artificial intelligence with cybersecurity practices to develop intelligent, autonomous systems capable of identifying and mitigating security threats efficiently.