Project HyperAdapt: An Agent-based intelligent sandbox design to deceive and analyze sophisticated malware

Shamalka Perera
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it21261046@my.sliit.lk

Shenuka Dias
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it21261664@my.sliit.lk

Vishwadinu Vithanage
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it21300950@my.sliit.lk

Avishka Dilhara
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it21299452@my.sliit.lk

Amila Senarathne
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
amila.n@sliit.lk

Deemantha Siriwardana
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
deemantha.s@sliit.lk

***Abstract*— Malware increasingly employs sophisticated evasion techniques to bypass sandbox-based analysis, rendering traditional detection methods ineffective. This research presents Project HyperAdapt: Agent-Based Intelligent Sandbox, a framework that integrates both offensive and defensive machine learning models to enhance malware detection, deception, and behavioral analysis. The offensive RL model generates evasive malware samples, challenging the sandbox, while the defensive models including hybrid evasion detection, GAN-based behavior simulation, and a dynamically adapting RL agent work collectively to improve sandbox resilience. By continuously learning from evasive malware behavior, the defensive RL agent adapts in real-time, strengthening detection capabilities. Experimental results demonstrate that this approach enhances sandbox effectiveness, ensuring long-term adaptability against evolving malware threats**.

Keywords— Malware Evasion, Malware Detection, Reinforcement Learning, , Dynamic Malware Analysis, GAN-Based Behavior Simulation, Hybrid Malware Detection, Adaptive Sandboxing

I. INTRODUCTION

The rapid evolution of malware presents a significant challenge to cybersecurity, as adversaries continuously develop sophisticated techniques to evade detection and analysis. Traditional static and signature-based detection methods struggle to keep pace with these advancements, leading to the widespread adoption of dynamic analysis systems such as sandboxing. Sandboxes provide an isolated environment to execute and observe malware behavior, offering deeper insights into its functionality. However, as malware authors recognize the presence of these controlled environments, they employ evasion tactics designed to detect, bypass, or deceive sandboxes. These techniques range from environmental checks and timing delays to behavioral alterations based on the absence of human interaction, significantly reducing the effectiveness of traditional sandbox-based detection.

Addressing the growing sophistication of malware evasion requires a dual approach that incorporates both offensive and defensive methodologies. From an offensive perspective, understanding how malware evades detection allows security researchers to anticipate and counter future threats. By actively developing evasion sequences and testing those new evasion sequences, it becomes possible to identify weaknesses in existing sandbox environments, providing critical insights into improving detection mechanisms. This approach not only strengthens malware research but also contributes to the development of more robust and resilient security solutions.

Conversely, the defensive perspective focuses on enhancing the ability of sandboxes to detect, analyze, and counter evasive malware. Modern cybersecurity strategies must extend beyond traditional detection techniques, incorporating intelligent automation, machine learning, and deception-based mechanisms to overcome sophisticated malware behavior. By improving sandbox environments to actively deceive and interact with evasive malware, security researchers can force malicious software to reveal its true functionality. A well-integrated defensive approach ensures that malware analysis remains effective, even as adversaries refine their evasion techniques.

To address these critical security challenges, we present Project HyperAdapt: Agent-Based Intelligent Sandbox**,** a comprehensive framework designed to enhance the detection, deception, and analysis of evasive malware. Project HyperAdapt integrates four distinct yet complementary components, each targeting specific weaknesses in traditional sandbox environments. By combining offensive and defensive strategies, this framework strengthens malware analysis by proactively identifying evasion techniques, improving detection accuracy, and enhancing sandbox resilience against sophisticated threats.

A. Offensive Reinforcement Learning Model:

This component focuses on developing a DQN based Reinforcement Learning malware evasion model that modifies malware dynamically to bypass Cuckoo Sandbox detection. The model begins with static evasion techniques like timing delays, human like behaviors, File system evasions and PE header modifications and progressively learns to apply dynamic static evasion sequences to improve evasion effectiveness. The system relies on two-way communication between RL Model and Cuckoo environment to automate malware modification, execution, and iterative learning. The primary objective is to train the RL agent to discover effective evasion strategies for Trojan and Ransomware families through continuous adaptation and testing.

B. User behaviour simulation within the sandbox:

This component enhances sandbox environments by automating realistic user behavior to counter sandbox-evasive malware. It leverages GAN (Generative Adversarial Networks) generated user profiles and executes timed interactions to create an authentic environment. This approach deceives malware into believing it is running on a real system instead of a sandbox.

C. Hybrid detection of sandbox evasion techniques:

This component is a hybrid detection system designed to identify and classify sandbox evasion techniques in Cuckoo Sandbox reports. By combining rule-based feature extraction with machine learning classification, This enhances malware analysis by automating the detection of evasive behaviors. The rule-based module scans sandbox logs for known evasion indicators, such as timing delays, anti-debugging API calls, and user interaction checks, while the machine learning model generalizes beyond predefined rules to detect previously unseen evasions. This hybrid approach ensures scalable, accurate, and adaptive evasion detection, significantly improving sandbox-based malware analysis and reducing the need for manual log inspection.

D. Dynamically modifying sandbox environment:

This component focuses on applying modifications to the sandbox environment dynamically and train the RL agent with the ideal action that must be taken to prevent the malware sample from hiding its malicious behaviour. The rewards or penalties for each action will be allocated with the comparison of Cuckoo's analysis score which is generated before applying an action and after applying an action.

The necessity of combining both offensive and defensive research within the same study is evident. Offensive strategies enable the proactive identification of emerging evasion methods, while defensive enhancements ensure that security mechanisms evolve accordingly. This research presents a comprehensive agent-based intelligent sandbox design that integrates both perspectives to improve malware analysis. The study is structured into distinct components, beginning with reinforcement learning-driven malware evasion, followed by hybrid evasion detection, synthetic user behavior simulation, and reinforcement learning for dynamic malware interaction. Each component addresses a specific challenge in malware evasion and sandbox security, collectively contributing to a more adaptive and intelligent malware analysis framework.

II. LITERATURE REVIEW

Sandbox-based malware analysis is a crucial method for detecting and studying malicious software, yet modern malware increasingly employs sophisticated evasion techniques to bypass detection. Existing research has explored static and dynamic analysis methods, but many sandbox solutions rely on predefined rules, making them ineffective against evolving threats. To address these limitations, Project HyperAdapt: Agent-Based Intelligent Sandbox integrates offensive and defensive strategies, leveraging reinforcement learning, machine learning, and deception techniques to enhance malware detection, deception, and behavioral analysis.

A. Offensive RL agent

Malware evasion techniques have evolved from basic virtualization checks [1] to sophisticated timing delays, API-hook evasion, and user-interaction mimicry [2]. Early malware used CPU and BIOS identifier checks to detect sandbox environments, but modern sandboxes obscure VM artifacts to counter these methods [3]. Trojans use multi-layered sandbox detection**,** analyzing system process lists, user activity, and environmental factors [4]. Ransomware employs execution delays and human-input triggers to avoid automated detection [5]. Prior Reinforcement Learning (RL)-based evasion models primarily focus on static file modifications [6] but lack structured evasion sequencing and real-time sandbox adaptation [7]. This research introduces a Deep Q-Network (DQN)-based RL model that dynamically learns evasion sequences, integrating two-way feedback from Cuckoo Sandbox. Unlike existing approaches, it directly learns from Cuckoo reports, optimizing both static and behavioral evasion techniques. By adapting to evolving defenses and specializing in Trojans and Ransomware, this research fills critical gaps in malware-specific sandbox evasion strategies**.**

B. User behaviour simulation within the sandbox

Traditional sandbox-based malware analysis systems often fail against sandbox-evasive malware, which detects artificial environments by checking for a lack of human-like activity[8]. Studies have shown that malware frequently monitors user interactions, such as mouse movements, keystrokes, file operations, and browsing behavior, to determine if it is executing in a real system or a sandbox environment [8][9].

To counter this, research has explored behavioral deception techniques that simulate user activity. Prior works have introduced scripted automation and replayed user interactions, but these methods often lack randomness and adaptability, making them easily detectable by advanced malware [8][9].

Our approach builds upon these methods by utilizing Generative Adversarial Networks (GANs) to generate user profiles and automate dynamic user behavior. It executes timed and adaptive interactions, making it harder for malware to distinguish between a sandbox and a real system. By incorporating randomness and variability, this method enhances the effectiveness of sandbox-based malware analysis.

C. Hybrid detection of sandbox evasion techniques

Malware that can evade sandbox environments poses a serious challenge in malware analysis. While there are methods to detect evasive malware after execution, there aren't many proactive approaches that can identify how malware avoids detection in the first place. Some existing research focuses on rule-based detection, which looks for patterns like delayed execution, sandbox detection checks**,** and user activity monitoring [10]. However, these techniques struggle with new or modified evasions. Machine learning-based malware detection has gained attention, with studies using API sequence analysis and behavior classification to distinguish malware from normal programs [11][12]. Still, these approaches do not classify evasion techniques directly from sandbox logs. While machine learning can improve detection, it requires large labeled datasets, which are scarce for sandbox evasion tactics [10][13]. To address this gap, this

research proposes a hybrid detection system that combines rule-based feature extraction with machine learning classification, creating an automated and adaptive method for detecting sandbox evasion techniques more effectively.

D. Dynamically modifying the sandbox environment

In the area of malware analysis, malwares' capability to outsmart sandbox environments is a topic that has a noble attention. Even though, there are reactive approaches available to detect malware samples that can hide their malicious behaviour inside a sandbox environment, proactive approaches are not widely available. Mills et al. [14], introduced some methods related to anti-evasion testing of malware. These methods belong to the reactive approach as they do not modify a sandbox environment dynamically. Several studies [15], [16], [17], [18] have suggested to detect malware using machine learning and deep learning techniques but none of these studies include any mechanism to modify a sandbox environment.

III. METHODOLOGY

The Agent-Based Intelligent Sandbox framework is a comprehensive security system, designed to address the challenges posed by sandbox-aware malware**.** This section details the methodologies employed in each of the four components of the agent, which collectively enhance the evasion detection, deception, and dynamic analysis capabilities of sandbox environments against sophisticated adversarial threats.

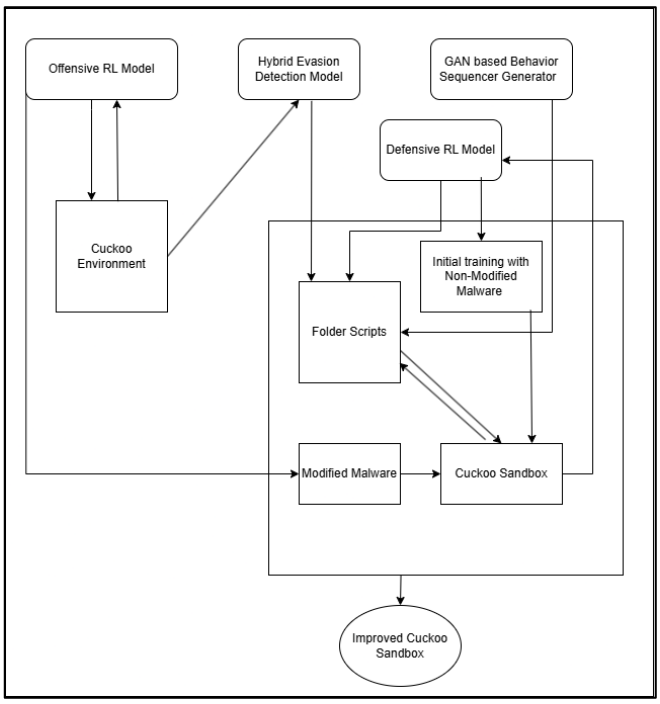

Figure 1: High-level system diagram

A. Offensive RL model

The Offensive DQN Reinforcement Learning (RL) model is a novel approach, in a structured way to bypass Cuckoo Sandbox Environment by turning static evasions to dynamic evasions sequences and testing those evasions against live locally installed cuckoo. The RL state population happened along with training process. The following is main steps of that mentioned structured approach,

Two-Way Communication with Cuckoo Sandbox (Real-Time Feedback Loop): This research establishes **a** two-way communication channel between the RL model and Cuckoo Sandbox**,** enabling real-time feedback integration**.** Each modified malware sample is executed in Cuckoo, which returns detection scores, behavioral analysis, and execution logs**.** The RL model processes this data to evaluate the effectiveness of evasion techniques and refines its strategy accordingly**.** Communication is implemented through SSH Reverse Tunneling or API-based interaction, ensuring a seamless data transfer. This dynamic feedback loop enables the RL model to iteratively improve evasion sequences based on actual sandbox responses.

Structured RL-Driven Evasion Sequence Optimization: A major limitation in existing RL-based malware evasion techniques is their reliance on single-action modifications or randomized adversarial changes without structured planning. This research addresses this issue by introducing structured evasion sequences, where the RL model selects and optimizes a sequence of evasion techniques from a predefined action table. The action table itself contains only individual evasion techniques such as time delays, human-like behavior simulations, file system evasions, and PE header modifications. However, the RL model autonomously determines the optimal evasion sequence, combining multiple techniques in a structured and adaptive manner. The model keeps track of successful and unsuccessful evasion sequences, avoiding ineffective combinations while reinforcing the most successful strategies. By optimizing a dynamic sequence of evasions, the RL model ensures higher stealth and adaptability, making malware evasion more effective.

Automated RL-Based Scoring and Self-Improvement System: A key aspect of this research is the direct integration of Cuckoo Sandbox feedback into the RL model's reward mechanism, ensuring continuous self-improvement. The RL model assigns rewards based on Cuckoo's detection score, dynamically linking malware performance to sandbox responses. If the detection score remains high, a strong negative reward is applied, discouraging ineffective evasion sequences. If the score remains unchanged, a small negative reward prevents stagnation. Conversely, if the score decreases, indicating successful evasion, the model assigns a positive reward, reinforcing the effective evasion sequence. This iterative learning process allows the RL model to continuously refine its strategies, tracking unsuccessful combinations while strengthening those that successfully bypass detection

B. User behaviour generation use GAN model

This part implements an automated user behavior simulation system to counter sandbox-evasive malware, making it harder for malware to detect artificial environments. The approach consists of three core components that work together to enhance the realism of sandbox environments.

User profile generation: A Generative Adversarial Network (GAN) is used to generate user profiles that encapsulate diverse characteristics, including mouse movements, typing speed, scrolling behavior, and tab-switching frequency. These profiles ensure that the sandbox exhibits human-like behavior, making it harder for malware to detect anomalies. Additionally, user interests are randomly

selected from a large pool, and Large Language Models (LLMs) determine relevant search queries and visited URLs based on these interests. This process creates a behavioral blueprint that guides user interactions throughout the sandbox session.

Automated user interaction execution: The system automates realistic user behaviors such as browsing activity, application usage, and document interactions to enhance authenticity. Search queries and visited URLs from the user profile dynamically shape browsing history, reinforcing the illusion of a real user. Timed and randomized, interactions prevent behavioral patterns from being easily detected.

Sandbox integration: All generated behaviors are deployed within Cuckoo Sandbox, where they execute autonomously according to a timetable-driven framework.

C. Hybrid Detection of Sandbox Evasion Techniques

This component presents a hybrid detection framework that integrates rule-based feature extraction with machine learning classification to systematically detect sandbox evasion techniques. The methodology is structured into three stages: rule-based feature extraction, machine learning-based classification, and hybrid detection integration**.**

Rule-Based feature extraction: The first stage extracts evasion-related features from sandbox execution logs using predefined detection rules. The system analyzes behavioral indicators associated with timing, human behavior, and anti-debugging evasions. These categories encompass techniques designed to delay execution, verify user presence, and detect analysis environments. Behavioral patterns indicative of evasions are identified and recorded, forming the feature set for further classification. This process ensures that known evasive behaviors are detected efficiently without requiring prior knowledge of specific malware families. The extracted features are structured into a dataset for subsequent processing.

Machine learning based classification: To enhance detection beyond predefined rules, a machine learning-based classification model is applied. The extracted behavioral features are transformed into structured numerical representations to facilitate pattern recognition. The classification model is trained on labeled execution logs, enabling it to distinguish between evasive and non-evasive behavior. The system generalizes detection by identifying previously unseen or modified evasions, addressing the limitations of static rule-based methods. The classifier is optimized for high precision to minimize false positives while maintaining adaptability to emerging evasion techniques.

Hybrid detection integration: The final stage integrates rule-based and machine learning-based detection in a sequential classification pipeline. Initially, the system applies rule-based detection to flag evasions that conform to predefined patterns. If no evasion is identified, the sample is processed by the machine learning model, which evaluates behavioral attributes to detect evasive activity. This hybrid approach ensures efficient detection of known techniques while maintaining adaptability for unknown or evolving evasion tactics. The final output consists of a structured classification report, detailing detected evasions with confidence scores. This report supports automated sandbox-based malware analysis, reducing the need for manual log inspection and enhancing detection accuracy. The combined approach enables scalable, adaptive evasion detection, improving resilience against evolving malware threats.

D. RL agent that dynamically modify sandbox environment

Malware analysis can be enhanced by integrating reinforcement learning (RL) with dynamic analysis tools like Cuckoo Sandbox. This approach automates decision-making processes, optimizing investigative actions for better detection and classification of malware. The system consists of key components, including a dynamic execution environment, an RL agent for decision optimization, custom action scripts for investigative tasks, feature extraction modules for structured data conversion, and a reward function to guide the agent's learning. Its modular architecture ensures adaptability and scalability, enabling continuous improvement in identifying malware threats.

Data Collection & feature engineering: Malware and benign samples are gathered from sources like VirusTotal and MalwareBazaar. Each sample is executed in a sandboxed environment, generating logs of system calls, file operations, and network activity. Relevant behavioral features such as API call sequences and registry modifications are extracted and structured as inputs for the RL model, ensuring meaningful real-world learning.

Reinforcement Learning for malware detection: The RL model, implemented using Deep Q-Networks (DQN) or Proximal Policy Optimization (PPO), learns malware behavior patterns iteratively. The state-action space maps system states investigative actions like API tracing or network inspection. A reward mechanism encourages effective actions and penalizes inefficiencies, refining the model's decision-making to detect previously unknown malware behaviors.

Evaluation & continuous improvement: Performance is assessed through detection accuracy, false positive/negative rates, exploration efficiency, and execution overhead. The RL-enhanced system is compared with traditional sandbox analysis to measure improvements in classification and resource utilization. Once validated, the model integrates into an automated pipeline for real-time threat detection, continuously adapting to evolving malware tactics through self-learning mechanisms.

IV. Results

The primary objective of this research is to develop a dynamically adapting sandbox defensive RL agent that enhances malware analysis by continuously evolving to counter advanced evasion techniques. While the offensive RL model generates evasive malware samples to test and challenge the sandbox, the three defensive models hybrid evasion detection, GAN-based behavior simulation, and the defensive RL model work collectively to improve the sandbox's resilience. Each component contributes to the overall goal by identifying evasive techniques, simulating realistic user interactions to deceive malware, and refining the sandbox's defensive strategies through reinforcement learning. The results obtained from these individual models provide valuable insights that drive the adaptation process, ensuring that the sandbox remains effective against evolving threats.

A. *Performance Metrics of the RL-Based Malware Evasion Model*

The RL-based malware evasion model significantly improves evasion effectiveness by dynamically refining structured evasion sequences. The Cuckoo detection score reduction (+0.2 to +2.5 lower) is achieved through optimized file system modifications, execution delays, and human-like behavior simulations, making malware harder to detect. The evasion success rate (60–80%) reflects the model's ability to learn and apply adaptive bypass strategies, while the false positive rate (2–5%) remains low due to optimized modifications preserving functional integrity. The model reduces per-sample analysis time (3–10 min) by automating evasion selection and removing redundant modifications, improving efficiency. The unique evasion discovery rate (80–95%) highlights its ability to generate new evasion techniques, while script execution time (5–10 min) is minimized through faster, structured evasion attempts. These improvements confirm that RL-driven malware evasion is adaptive, efficient, and stealth-focused, dynamically optimizing evasion sequences based on real-time sandbox feedback.

Table 1: Performance Metrics of Offensive RL Model

Metric	Baseline	With RL
Cuckoo Detection Score	General score	0.2 to +2.5 reduction
Evasion Success Rate (%)	20–40	60–80
False Positive Rate (%)	3–7	2–5
Per-Sample Analysis Time (min)	5–15	3–10
Unique Evasion Discovery Rate (%)	50–70	80–95
Script Execution Time (min)	10–15	5–10

B. *Evaluation of GAN model that generates user profiles*

To evaluate the accuracy of the GAN-generated user behavior**,** several statistical metrics were used to compare the synthetic data with real-world user interactions.

KSComplement (Mean): Measures how closely the generated behavior follows real-world user behavior distribution (Kolmogorov-Smirnov test). Higher values indicate better accuracy.

TVComplement (Mean): Represents the Total Variation distance between real and generated data. Values close to 1 suggest high similarity.

Correlation Similarity: Indicates statistical correlation between synthetic and real user behavior patterns. A value close to 1 shows strong alignment.

Categorical Coverage (Mean): Measures whether all possible categories of user behavior are represented in the generated dataset. A value near 1 indicates comprehensive coverage.

Table 2: Performance Metrics of GAN Model

Metric	Value
KSComplement (Mean)	0.8396
TVComplement (Mean)	0.8667
Correlation Similarity	0.9783
Categorical Coverage (Mean)	0.9995

C. *Evaluation of Evasion Hybrid Detection system*

The proposed hybrid detection system was evaluated for its ability to accurately detect and classify sandbox evasion techniques using Cuckoo Sandbox reports. The results demonstrate that the system effectively identifies evasive behaviors while maintaining high detection accuracy and minimal false positives.

Detection Accuracy**:** The hybrid system achieved a 95% accuracy rate in detecting and classifying evasion techniques. The integration of rule-based detection and machine learning classification allowed the system to generalize beyond predefined rules, improving detection of previously unseen evasions**.**

Performance Comparison**:** Compared to standalone rule-based and machine learning-based methods, the hybrid system provided higher precision and recall while maintaining efficient detection speed. The rule-based approach effectively identified known evasions but struggled with new variants, while the ML classifier improved adaptability but required more computational resources. The hybrid model successfully combined interpretability with adaptability, optimizing both performance and accuracy.

Table 3: Performance Metrics of Hybrid Evasion Detection System

Detection Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Rule-Based Only	85	90	75	82
ML-Based Only	92	95	88	91
Hybrid Approach	95	97	92	94

D. *Improved detection results of dynamically modified sandbox*

These factors have been chosen because they collectively determine the effectiveness, reliability, and efficiency of a malware detection system. Malware Detection Rate is crucial as it measures the system's ability to correctly identify threats, ensuring strong protection. However, attackers use sophisticated techniques to evade detection, making the Evasion Detection Rate vital to assess how well the system handles stealthy malware. Execution Path Coverage is included because malware often behaves differently based on execution conditions, and thorough analysis improves detection accuracy. False Positive Rate matters because misclassifying benign files can disrupt operations and reduce trust in the system. Lastly, Per-Sample Analysis Time is important since faster detection minimizes the impact of malware, balancing speed with thorough security assessment.

Table 4: Performance Metrics of Adpative Sandbox

Metric	Baseline	With RL	Expected Improvement
Malware Detection Rate	70–85%	85–95%	+10–20%
Evasion Detection Rate	20–40% missed	5–15% missed	-25–35%
Execution Path Coverage	40–60%	70–90%	+30–50%
False Positive Rate	3–7%	2–5%	-1–2%
Per-Sample Analysis Time	5–15 min	3–10 min	-30%

V. Future Research directions

While Project HyperAdapt makes a significate step forward, there are several avenues for future research.

Adaptive evasion strategies with advanced RL architecutures: To further enhance offensive RL-driven malware evasion, the research can evolve by expanding evasion techniques, refining RL architectures, and targeting broader malware families. Current work focuses on static evasion categories, but integrating polymorphic and metamorphic evasion techniques, memory-resident execution, and process injection could further reduce detection rates. Additionally, advanced RL architecture such as Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), or Multi-Agent RL could improve policy adaptation and decision-making efficiency, allowing the model to dynamically adjust evasion strategies based on sandbox behavior changes. Incorporating adaptive learning against multiple sandbox environments (e.g., FireEye, ANY.RUN, or Drakvuf) would make evasion sequences more robust across different detection mechanisms. Expanding to Rootkits, Fileless Malware, and APT**s** would also broaden attack scenarios, enabling the model to learn evasive behaviors for more sophisticated malware families.

A real-time user behaviour adjustment mechanism: As a future research direction, a mechanism can be developed to enhance sandbox effectiveness by dynamically adjusting simulated user behavior in response to malware interactions. By detecting evasion attempts, the system can trigger adaptive user actions, such as mouse movements and window switches, creating a more realistic environment that makes it harder for malware to distinguish the sandbox from a real system.

New evasion technique variations: Future research could focus on the generation of new evasion technique variations to improve proactive detection and enhance the adaptability of sandbox analysis. This would involve automatically generating modified evasion strategies by altering existing behaviors or combining multiple evasion tactics to create more sophisticated variations.

Enhancing Action-Selection Strategies and Implementing Advanced RL Techniques: For future work, the integration of reinforcement learning (RL) with Cuckoo Sandbox can be further enhanced by refining the action-selection strategies to improve malware behavior exploration. Implementing deep RL techniques, such as Proximal Policy Optimization (PPO) or Deep Q-Networks (DQN), could enhance the adaptability of the agent in identifying novel malware evasive techniques. Additionally, incorporating automated feature extraction from system call logs and behavioral reports could improve model efficiency and detection accuracy. Expanding the research to cover obfuscated and polymorphic malware variants would also strengthen the robustness of the approach. Lastly, integrating real-time threat intelligence feeds could enable dynamic policy updates, making the system more responsive to emerging threats.

VI. Conclusion

This research presents Project HyperAdapt: Agent-Based Intelligent Sandbox, a dynamically adapting malware analysis framework that integrates offensive and defensive machine learning techniques to counter evasive malware. By combining reinforcement learning-based malware evasion, hybrid evasion detection, adversarial user behavior simulation, and an adaptive defensive RL agent, the system enhances sandbox resilience against evolving threats. Experimental results demonstrate that this approach significantly improves malware detection, deception, and behavioral analysis, allowing the sandbox to continuously adapt to new evasion strategies. The findings highlight the effectiveness of offensive defense integration in malware research, paving the way for future advancements in intelligent, self-learning security systems.

References

[1] S. Ž. Ilić, M. J. Gnjatović, B. M. Popović, and N. D. Maček, "A Pilot Comparative Analysis of the Cuckoo and Drakvuf Sandboxes: An End-User Perspective," Vojnotehnički Glasnik / Military Technical Courier, vol. 70, no. 2, pp. 372-392, 2022, doi: 10.5937/vojtehg70-36196.

[2] A. A. R. Melvin and G. J. W. Kathrine, "A Quest for Best: A Detailed Comparison Between Drakvuf VMI-Based and Cuckoo Sandbox-

Based Techniques for Dynamic Malware Analysis," in Intelligence in Big Data Technologies - Beyond the Hype, Singapore: Springer, 2020, pp. 386-395, doi: 10.1007/978-981-15-5285-4_27.

[3] T. Quertier, B. Marais, S. Morucci, and B. Fournel, "MERLIN: Malware Evasion with Reinforcement Learning," arXiv preprint arXiv:2203.12980v4, Mar. 2022.

[4] W. Song, X. Li, S. Afroz, D. Garg, D. Kuznetsov, and H. Yin, "MAB-Malware: A Reinforcement Learning Framework for Attacking Static Malware Classifiers," arXiv preprint arXiv:2003.03100v3, Apr. 2021..

[5] R. Labaca-Castro, S. Franz, and G. D. Rodosek, "AIMED-RL: Exploring Adversarial Malware Examples with Reinforcement Learning," in Proc. 16th International Conference on Availability, Reliability and Security (ARES '21), Vienna, Austria, 2021, pp. 1-9,

[6] X. Li and Q. Li, "An IRL-based malware adversarial generation method to evade anti-malware engines," Computers & Security, vol. 104, pp. 102118, 2021, doi: 10.1016/j.cose.2020.102118.

[7] W. Song, X. Li, S. Afroz, D. Garg, D. Kuznetsov, and H. Yin, "MAB-Malware: A Reinforcement Learning Framework for Blackbox Generation of Adversarial Malware," in Proc. 2022 ACM Asia Conference on Computer and Communications Security (ASIA CCS

[8] S. Liu, P. Feng, S. Wang, K. Sun, and J. Cao, “Enhancing malware analysis sandboxes with emulated user behavior,” Comput. Secur., vol. 115, no. 102613, p. 102613, 2022.

[9] L. Wang et al., “User behavior simulation with large language model based agents,” arXiv [cs.IR], 2023.

[10] C. Herzog, V. Tong, P. Wilke, A. Van Straaten, and J.-L. Lanet, “Evasive Windows Malware: Impact on Antiviruses and Possible Applications, vol. 103, pp. 249–261, Feb. 2018, doi: https://doi.org/10.1016/j.jnca.2017.10.004.

[11] Ananya Redhu, P. Choudhary, K. Srinivasan, and Tapan Kumar Das, “Deep learning-powered malware detection in cyberspace: a contemporary review,” Frontiers in physics, vol. 12, Mar. 2024, doi: https://doi.org/10.3389/fphy.2024.1349463.

[12] A. Mills and P. Legg, “Investigating anti-evasion malware triggers using automated sandbox reconfiguration techniques,”J. Cybersecur. Priv., vol. 1, no. 1, pp. 19–39, 2020.

[13] S. Zhang, M. Gao, L. Wang, S. Xu, W. Shao, and R. Kuang, “A malware-detection method using deep learning to fully extract API sequence features,”Electronics (Basel), vol. 14, no. 1, p. 167, 2025.

[14] B. Bokolo, R. Jinad, and Q. Liu, “A Comparison Study to Detect Malware using Deep Learning and Machine learning Techniques,” in2023 IEEE 6th International Conference on Big Data and Artificial Intelligence (BDAI), 2023, pp. 1–6.

[15] Countermeasures,” Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, 2020, doi: https://doi.org/10.5220/0009816703020309.

[16] L. Shiva, M. A. Ajay Kumara, and C. D. Jaidhar, “Windows malware detection based on cuckoo sandbox generated report using machine learning algorithm,” International Conference on Industrial and Information Systems, Dec. 2016, doi: https://doi.org/10.1109/iciinfs.2016.8262998.

[17] M. Noor, H. Abbas, and W. B. Shahid, “Countering cyber threats for industrial applications: An automated approach for malware evasion detection and analysis,” Journal of Network and Computer