IRJCS: International Research Journal of Computer Science
Volume 13, Issue 01, January 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1301.06

Red Team AI Malware Simulator: Dual-Phase AI-Driven Polymorphic Malware Testing Framework

Dr.Rachitha M V
Associate Professor, Dept. of CSE
Vemana Institute of Technology, Bengaluru, India
rachitha.mv@vemanait.edu.in
https://orcid.org/0000-0002-9537-8158
Muhammed Abdul Muid, Rohan Reddy, Santhosh Kumar S, Vishwas S
Student, Department of CSE,
Vemana Institute of Technology, Bengaluru, India

Publication History
Manuscript Reference: IRJCS/RS/Vol.13/Issue01/CSJA26.JACS10085
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.13/Issue01/CSJA26.JACS10085
Received:12,December 2025,Revised:24,December 2025,Accepted:02 January 2026 Published Online:20 January 2026
https://www.irjcs.com/volumes/Vol13/iss-01/06.CSJA26.JACS10085.pdf
Article Citation:Dr.Rachitha,Muhammed,Rohan,Santhosh,Vishwas(2026),Red Team AI Malware Simulator: Dual-Phase AI-Driven Polymorphic Malware Testing Framework, IRJCS: International Research Journal of Computer Science, Volume 13, Issue 01 of 2026 pages 29-32
Doi:>https://doi.org/10.26562/irjcs.2026.v1301.06
BibTeX Key Dr.Rachitha@2026Red
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2026, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1301.06 The journal's official abbreviation is IRJCS.
Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: With the rise of artificial intelligence, the cyber security landscape is witnessing a paradigm shift where traditional penetration testing tools that rely on static attack patterns and obfuscation are increasingly ineffective against modern AI- driven security defenses. This work presents the RED TEAM AI MALWARE SIMULATOR, a dual-phase AI-driven polymorphic malware testing framework with comprehensive C2 orchestration and intelligent decision making. The framework uses a simple two-phase approach consisting of reconnaissance and attack, driven by custom-trained Recon Prioritization and Attack Decision AI models integrated with a bespoke Django-based C2 server deployed on a VPS. The system simulates sophisticated, real- world malware behavior in a controlled and legal environment, including full kill chain simulation from reconnaissance to C2 communication, while ensuring that all executed actions are safe, non-destructive, and sand boxed. Experimental evaluation consists of five comprehensive tests coverings can submission, recon prioritization, attack decision, report generation, and LLM mitigation, with all five tests passing (100% success rate), demonstrating that modern systems are vulnerable to AI-powered malware and highlighting the need for AI-aware security measures in organizations.
Index Terms: AI-powered malware, polymorphic malware simulation, command and control server, recon prioritization, attack decision model, LLM mitigation, red team tooling

I. INTRODUCTION

The rapid advancement of artificial intelligence has fundamentally altered the cyber security domain, rendering many traditional defense mechanisms obsolete [1]. Traditional security measures are becoming increasingly vulnerable to AI-assisted threats that can autonomously adapt and evade detection [2]. Penetration testing tools are used by cyber security professionals to test the security of enterprises and their networks. With the recent boom in AI and graphical processing units getting more powerful each generation, which can be used to train more powerful models, one must consider how hackers can exploit agent AI to attack networks. Traditional penetration testing tools rely on static attack patterns and obfuscation, making them ineffective against modern AI-driven security defenses. AI-powered malware can adapt dynamically, bypassing firewalls, EDR systems, and behavioral analysis. Current research lacks a comprehensive framework to study its full attack lifecycle. There is a lack of public research in this domain, leaving in dependent developers and security researchers without any means of protection against AI-powered malware attacks. This work presents the Red Team AI Malware Simulator, a dual-phase AI-driven polymorphic malware testing framework. The system simulates sophisticated, real world malware behavior in a controlled environment, utilizing custom-trained Recon Prioritization and Attack Decision AI models orchestrated by a Django-based C2 server. The frame- work operates in two phases: System Reconnaissance and C2- Driven Attacker Simulation. In the first phase, the payload driver gathers system intelligence, which is prioritized by the Recon AI. In the second phase, the Attack Decision AI directs the payload to execute safe, simulated malicious actions. This approach allows red teams to evaluate system resilience against adaptive threats. Crucially, the simulation ensures safety by sandboxing actions and using an LLM mitigation module to prevent destructive behavior.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 01, January 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1301.06

A. Background and Literature Review

Prior work has explored artificial intelligence as a new vector for offensive security, evolving malware variants as antigens for antivirus systems, and camouflage in malware from encryption to metamorphism [3], as well as polymorphic execution platforms [4] and polymorphic attacks that deceive AI-based malware detection. These works report recon work- flow success rates of approximately 90%, exploit execution rates of approximately 50%, high evasion rates against large numbers of AV engines, and CNN-based detection models with high accuracy on both Linux and Windows systems [5], along with evasion rates up to 100% in some scenarios. Despite these advances, there is a lack of public research and tooling focused on a comprehensive framework to study the full attack life cycle of AI-powered malware in a controlled, legal setting targeted at red teamers and security professionals. The Red Team AI Malware Simulator addresses this gap by providing a dual-phase AI-driven polymorphic malware testing framework with bespoke C2 orchestration and integrated AI models for reconnaissance prioritization and attack decision making.

II. METHODOLOGY

The proposed Red Team AI Malware Simulator is a dual-phase framework comprising a C2 server, two AI models, and a payload driver, organized into a pipeline of Reconnaissance, Attack Simulation, and Report Generation.

A. System Architecture

The architecture features a Django-based C2 server deployed on a VPS, acting as the central orchestrator. It manages two-way communication with the payload driver (malware agent) deployed on the target Windows machine using JSON format for data exchange. The system includes two custom AI models: the Recon Prioritization Model and the Attack Decision Model. The payload driver, built in C/C++and Rust for lower-level system access with Python integration for orchestration, executes commands and gathers intelligence. An LLM mitigation module ensures safety by sanitizing actions and preventing execution of unsafe or real destructive commands.

B. AI Models

The Recon Prioritization AI Model is a custom neural network that ranks discovered files based on sensitivity using file metadata and location. It inputs raw recon data (file system data, metadata, names, locations, types) and outputs a prioritized list to simulate experienced red-team focus. It assigns priority scores to targets based on sensitivity, accessibility, and value.

The Attack Decision AI Model predicts the optimal next attack action based on the current environment state, system configurations, and recon results. It inputs current environment data, available vulnerabilities, defense mechanisms detected, and prioritized recon results. It outputs the next best course of action for the attack, maximizing attack efficiency while maintaining stealth. It adapts attack strategy based on real- time environment data.

C. Workflow

The workflow proceeds in two phases:

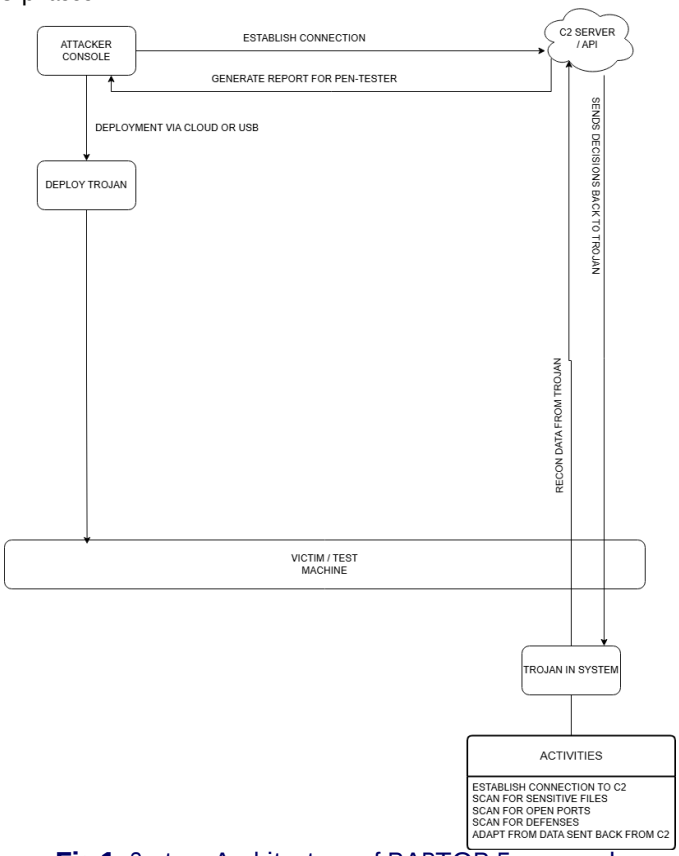

Fig.1. System Architecture of RAPTOR Framework

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 01, January 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1301.06

1) Reconnaissance: The payload driver scans the target (OS finger printing, port scanning, service enumeration, file system exploration) and sends data to the C2. The Recon AI prioritizes findings.
2) Attack Simulation: The C2 sends tasks based on the Attack Decision AI's decisions. The payload executes safe, simulated attacks (e.g., process listing, file reading, telemetry collection). Finally, the system generates a comprehensive PDF report and uses an LLM to provide actionable mitigation strategies.

III. RESULTS

The experimental evaluation focused on C2 functionality, AI model performance, and reporting quality. Five critical tests (scan submission, recon prioritization, attack decision, report generation, and LLM mitigation) were conducted, all achieving a 100% pass rate.

A. Recon Prioritization AI Model Performance

The Recon Prioritization AI model demonstrated strong performance in distinguishing between Sensitive and Not Sensitive classes. The classification report indicates high precision for the *Sensitive* class and high recall for the *Not Sensitive* class, with balanced F1-scores. Feature importance analysis revealed that file size is the most influential factor, followed by directory-based features. The precision-recall curve demonstrates that the model maintains high precision across a wide range of recall values, with an average precision of 0.95. The ROC curve shows an area under the curve (AUC) of 0.93, indicating strong overall discriminative ability.

B. Attack Decision AI Model Performance

The Attack Decision AI model results indicate robust multiclass classification performance across simulated attacker actions. The model consistently achieved high precision, recall, and F1-scores for each class, effectively distinguishing Between actions like process listing and file reading. Most classes of the automated reporting and LLM-based mitigation. The system highlights the vulnerability of modern defenses to adaptive AI threats. Future work will focus on extending support to Linux file systems, fine-tuning dedicated LLMs for mitigation, and expanding the range of simulated attack vectors. Exhibit F1-scores above 0.9. Feature importance analysis high- lighted reliance on features related to file sensitivity and previous action context. The model's decision process closely mimics an intelligent attacker, prioritizing actions based on the perceived value of discovered files and the sequence of prior steps.

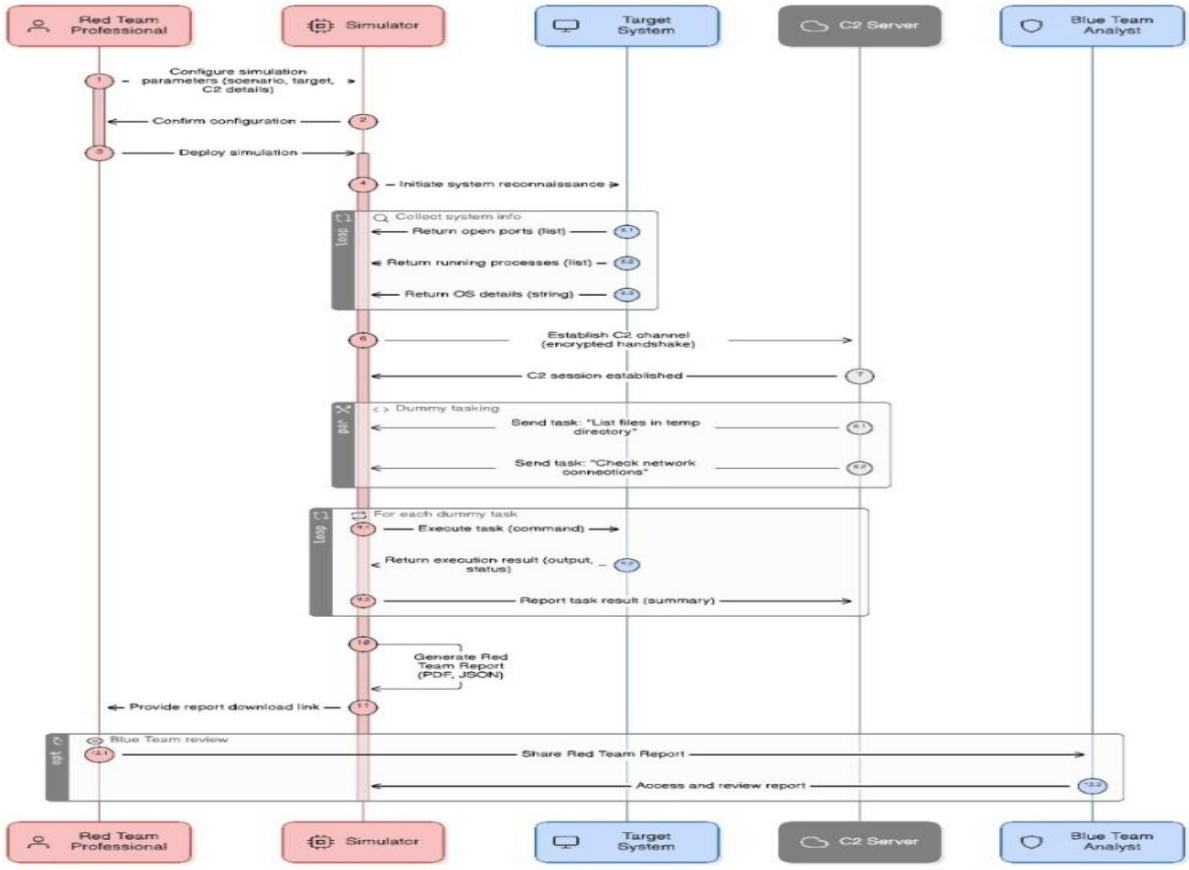

Fig.2. Operational Sequence Diagram

C. Overall System Performance

Report generation successfully produced comprehensive PDF documents containing all session data, while the LLM mitigation module provided specific, actionable security recommendations. The system demonstrated resilience and stealth in a controlled sandbox environment with active antivirus software [10].

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 01, January 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1301.06

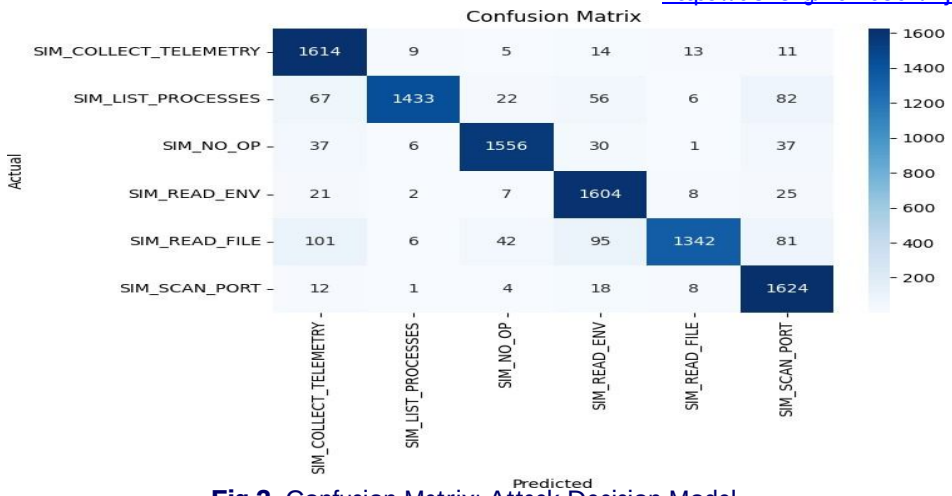

Fig.3. Confusion Matrix: Attack Decision Model

IV. CONCLUSION

The Red Team AI Malware Simulator successfully demonstrates a dual-phase, AI-driven framework for simulating sophisticated malware behavior in a controlled environment [9]. All five critical tests passed, validating the efficacy of the C2 orchestration, the accuracy of the Recon Prioritization and Attack Decision AI models, and the utility.

REFERENCES

1. L.J.Valencia, "Artificial Intelligence as the New Hacker: Developing Agents for Offensive Security," arXiv:2406.07561, 2024.
2. R.Murali, P.Thangavel, and C.S.Velayutham, "Evolving Malware Variants as Antigens for Antivirus Systems," Expert Systems with Applications, vol. 226, p. 120092, 2023. https://doi.org/10.1016/j.eswa.2023.120092
3. B.B.Rad,M.Masrom,and S.Ibrahim,"Camouflage in Malware: From Encryption to Metamorphism, "International Journal of Computer Science and Network Security (IJCSNS), vol. 12, no. 8, 2012.
4. A.Lohchab,"Investigating Polymorphism for the D-TIME Malware Execution Platform," Master's thesis, Masaryk University, Faculty of Informatics, Brno, Czech Republic, 2021.
5. C.Catalano, A.Chezzi, M.Angelelli, and F.Tommasi, "Deceiving AI-based Malware Detection through Polymorphic Attacks," Journal of Information Security and Applications, 2021. https://doi.org/10.1016/j.compind.2022.103751
6. W.Hu and Y.Tan, "Generating adversarial malware examples for black-box attacks based on GAN," arXiv preprint arXiv:1702.05983, 2017.[Online]. Available: https://arxiv.org/abs/1702.05983
7. B.Kolosnjaji,A.Demontis,B.Biggio,M.Maiorca,D.Giacinto,and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables," arXiv preprint arXiv:1803.04173,2018. [Online]. Available: https://arxiv.org/abs/1803.04173
8. M.Rigaki and S.Garcia,"Bringing a GAN to a knife-fight: Adapting malware communication to avoid detection,"IEEE Security and Privacy Workshops (SPW), 2018. [Online]. Available: https://arxiv.org/abs/1801.02629
9. IBM Research, "Deep Locker: Concealing targeted attacks with AI lock smithing, "Black Hat USA, 2018.[Online]. Available: https://research.ibm.com/publications/deeplocker-concealing-targeted-attacks-with-ai-locksmithing
10. Y.Liu,Z.Sun,andZ.Wu," Pentest GPT: An LLM-powered autonomous penetration testing agent,"arXiv preprint arXiv:2308.08439,2023.[On-line]. Available: https://arxiv.org/abs/2308.08439