IJARETY

International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 13, Issue 1, January - February 2026

Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 2, March-April 2026 ||

DOI:10.15680/IJARETY.2026.1302030

Multi-Agent Cybersecurity Red-Teaming Automation System

Vaibhav Chikkamath

Master of Computer Applications, CMR Institute of Technology, Bengaluru, India

ABSTRACT: As attacks against network, host, and application layers grow increasingly intricate, there is a pressing requirement for automated red-teaming tools capable of faithfully replicating genuine compromise scenarios. While progress in offensive automation has improved specific tasks like reconnaissance, vulnerability detection, and isolated exploits, contemporary systems rarely deliver fully verified, end-to-end attack chains. Historic approaches often face a trade-off, forcing a choice between frameworks that coordinate agents with restricted roles and planning tools that map out attack paths without actually deploying payloads. Consequently, the reliable automation of post-exploitation actions, privilege escalation, evidence collection, and final reporting remains largely unresolved.

This study investigates the current landscape of multi-agent red-teaming and breach-and attack simulations, evaluating their range, execution depth, and validation protocols. The analysis highlights a critical deficiency in systems that attempt to pair live exploitation with centralized planning while ensuring the preservation of verifiable proof-of-compromise. Furthermore, the paper details how orchestration rooted in actual execution allows planning logic to integrate with practical offensive tools, enabling reproducible, full spectrum assessments across diverse target environments. Ultimately, the results underscore the necessity of shifting research away from theoretical attack modelling toward validated, tool-centric automation that accurately mirrors real-world adversary behaviour.

KEYWORDS: Automated Red Teaming, Multi-Agent Systems, Cybersecurity Assessment, Full-Chain Exploitation, Breach and Attack Simulation, Vulnerability Validation, Penetration Testing Automation, Exploit Orchestration, Centralized Attack Planning.

I. INTRODUCTION

Assessing how resilient digital systems are against genuine offensive tactics typically relies on Red Teaming as a foundational practice. Security professionals have historically conducted these assessments manually, handling everything from the initial planning and reconnaissance phases to final exploitation and reporting. Although this approach offers realism, the workflows involved are often resource-heavy, constrained by time, and difficult to replicate consistently across different targets. With enterprise infrastructures now encompassing a mix of host environments, web applications, and network services, the intricacies of executing coordinated attacks have prompted researchers to investigate automated solutions for offensive security.

Developments in penetration testing automation have brought forward frameworks driven by agents, which allow reconnaissance and exploitation duties to be distributed across multiple components. Research into multi-agent solutions, such as PentestAgent and MAPTA, indicates that coordinating these agents can reduce the manual workload required for distinct tasks like reconnaissance, vulnerability detection, and targeted exploitation [1], [2]. Supporting this direction, other studies suggest collaborative agent systems designed to handle complex attack scenarios, such as discovering vulnerabilities across multiple surfaces or executing multi-step exploitations [3], [4]. These capabilities are further expanded by knowledge-augmented methods, which incorporate repository-stored offensive data to refine how exploits are selected and decisions are made [5].

Security environments based on reinforcement learning have also emerged, aiming to replicate attacker behaviour using adaptive strategies. Through tools such as CyberBattleSim and PenGym, agents undergo training for specific actions like lateral movement, privilege escalation, or picking exploits, utilizing reward-based mechanisms to learn [9], [11]. Researchers have also applied reinforcement learning to distinct offensive phases, specifically examining post-exploitation and shell persistence to help agents iteratively refine their exploitation tactics [13], [14]. Collectively, this body of work highlights an increasing shift toward creating autonomous offensive agents capable of learning tailored attack pathways.

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 2, March-April 2026 ||

DOI:10.15680/IJARETY.2026.1302030

Existing work in automated red teaming, however, shares two primary constraints. First, the majority of multi-agent systems limit their scope to a single environment—typically focusing solely on network scanning or web exploitation—which results in fragmented attack coverage rather than a comprehensive compromise workflow [1]–[4]. Second, environments that rely on reinforcement learning tend to be simulation-based, often halting at emulated attacks rather than deploying real payloads or verifying compromise against live system configurations [9], [10], [11]. Consequently, these automated systems frequently fail to provide verifiable proof-of-compromise or lack the adaptability needed to function outside restricted attack domains. Because of these shortcomings, there is a clear need for execution-driven automation that unifies planning, exploitation, privilege escalation, and evidence validation into a single, cohesive offensive process. Accordingly, this review examines the current state-of-the-art in multi-agent and reinforcement-learning red-teaming systems, assessing their execution depth, validation mechanisms, and attack coverage to identify pathways for scalable, evidence-based automation that mirrors true adversary behaviour across diverse targets.

II. LITERATURE REVIEW

2.1 Multi-Agent Automation in Penetration Testing

Work in offensive security automation has increasingly shifted toward coordinating multiple agents to reduce the need for manual intervention. These frameworks typically split up duties—such as scanning for vulnerabilities, reconnaissance, or running exploits— assigning them to distinct agents that operate within specific, limited domains. PentestAgent adopts this method by assigning separate roles for enumeration, finding vulnerabilities, and executing exploits [1]. In a similar vein, MAPTA streamlines the workflow for exploiting web applications by using specific agents to handle crawling, manipulating inputs, and injecting payloads [2]. Taking collaboration, a step further, VulnBot merges exploitation and reconnaissance agents into shared pipelines, allowing shared data to drive offensive moves against multiple targets [3]. These studies effectively demonstrate that multi-agent systems can lower manual workloads and boost automation levels in specific testing phases, though most remain restricted to a single surface or a narrow application field.

2.2 Knowledge-Enhanced and Collaborative Offensive Agents

Apart from just distributing tasks, newer studies focus on using stored offensive data and collaborative reasoning to improve how agents make decisions. To make attacks against various software configurations more reliable, xOffense utilizes a knowledge base to guide exploit selection [5]. Another approach, Shell or Nothing, employs agents activated in memory to benchmark realistic red teaming; these agents use derived insights to dynamically escalate privileges or stabilize shell access [6]. AutoRedTeamer presents collaborative agents that run for extended periods, continuously expanding their attack knowledge and refining concrete exploitation strategies [7]. While these improvements lead to better collaboration and more consistent exploits, they primarily depend on decentralized decision-making—where agents choose their own tasks instead of following a centralized planning controller.

2.3 Reinforcement Learning Approaches in Cyber Offense

Simultaneously, other research in cyber offense explores reinforcement learning techniques to automate decision-making during attacks. CyberBattleSim, for instance, models how attacks spread through networks using reward functions that encourage privilege escalation or greater access [9]. Similarly, PenGym creates environments where agents train via trial and error, learning to conduct reconnaissance, deploy exploits, or escalate privileges [11]. Other projects based on reinforcement learning target specific attack stages, such as automated privilege escalation or persistence and post-exploitation mechanisms like Raijū [12], [13]. Although these systems offer adaptive decision-making within simulations, they rarely execute against live infrastructure. Moreover, because they depend on simulated reward models and lengthy training cycles, reinforcement learning frameworks are not well-suited for practical, full-chain exploitation pipelines. Conversely, approaches that do not use reinforcement learning—specifically centralized planners managing execution agents—work directly with evidence-based compromise using valid actions and real tools, which better serves the practical requirements of enterprise penetration testing.

2.4 Planning-Based Simulation versus Real Execution

Another avenue of research is Breach and Attack Simulation (BAS), where planning logic automatically generates attack paths. Aurora illustrates this method, converting tool documentation into structured steps to create multi-stage attack sequences [3]. Although these systems offer robust modelling features, they typically avoid launching real payloads, stopping short at simulated compromise. Other planning research concentrates on formalizing test sequences or modelling penetration attack pathways [16], [19]. While these initiatives enhance reproducibility and map out attack routes effectively, they fail to validate exploit results on real systems or capture privilege states and shell evidence.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 2, March-April 2026 ||

DOI:10.15680/IJARETY.2026.1302030

Consequently, current planning-based solutions lack verifiable data on compromise, making their results difficult to apply in practical security assessments.

Summary Synthesis

The literature reveals a rapid move toward automating discrete red-teaming tasks. Multiagent methods help streamline exploitation and reconnaissance [1]–[5], whereas models driven by collaborative knowledge improve exploit reliability across different attacks [5]– [7]. Solutions using reinforcement learning introduce adaptability but generally remain restricted to simulated environments [9], [11]–[13]. While planning-based BAS tools differ in nature, excelling at generating structured attack paths, they typically stay within the realm of simulation rather than proceeding to validated exploitation [3], [16], [19]. Collectively, these comparisons highlight a shared deficiency across research efforts: very few existing systems manage to coordinate full-chain exploitation through centralized planning while executing real attacks and recording verifiable evidence of compromise on heterogeneous targets. Ideally, this gap warrants the development of execution-driven orchestration models designed to autonomously plan, execute, validate, and document real-world attacks using actual offensive tools—an approach that aligns with centralized agent coordination.

III. REVIEW-BASED METHODOLOGY

This segment outlines a review-based strategy for identifying existing offensive automation methods and establishing the methodological foundation for a centralized, execution-oriented penetration testing framework. Building upon cyber simulations driven by reinforcement learning and systems with partial agent orchestration, this review evaluates how multi-agent penetration testing functions within a planning-centric architecture designed to validate the entire attack chain.

3.1 Review Approach and Comparative Selection

The methodology relies on a comparative synthesis of multi-agent offensive frameworks, tools for execution-based penetration, and attack coordinators driven by knowledge bases. The selection process prioritized studies that:

- Automate distinct tasks such as reconnaissance, vulnerability scanning, or exploitation through specific agent roles.
- Execute actual payloads rather than relying on simulated attack modelling.
- Demonstrate operational workflows that are applicable across heterogeneous target environments.

Prominent examples, including PentestAgent, MAPTA, VulnBot, and xOffense, demonstrate the orchestration of reconnaissance and exploitation tasks via autonomous agents [1]– [5]. These works establish a comparative baseline for evaluating how different systems assign, coordinate, and validate agent responsibilities within an offensive workflow. Further research into knowledge-driven exploitation provides added perspective on agent collaboration, specifically regarding shared intelligence during multi-stage attacks [5], [6], [7].

Such categorization lays the methodological groundwork for execution-focused orchestration, emphasizing the necessity of coordinated direction over agent self-selection.

3.2 Attack-Workflow Automation Evaluation

Contemporary multi-agent studies demonstrate the automation of isolated attack phases, yet they exhibit significant divergence in their orchestration methods and workflow continuity. Frameworks such as MAPTA and PentestAgent depict structured transitions between reconnaissance, scanning, and exploitation. Nevertheless, their operations often remain confined to specific surfaces, such as web application inputs or standard network services [1], [2]. In contrast, VulnBot advances this model by embedding joint logic for reconnaissance and exploitation, adjusting agent behaviours as surface information evolves [3]. Likewise, xOffense demonstrates how knowledge-enhanced agent systems can select exploit strategies by leveraging stored attack intelligence [5].

A major limitation across most of these approaches, however, is the absence of global planning control. Agents typically determine actions autonomously based on their individual capabilities, resulting in a decentralized workflow. This decentralization constrains the consistency of full-chain attacks, particularly during transitions from exploitation to validation, privilege escalation, or reporting phases. As noted in distinct studies like Shell or Nothing, verification driven by evidence stays inconsistent when there is no explicit control directing an agent to validate its own output [6]. This reinforces the methodological requirement for a central decision-making system that governs agent execution across every phase, rather than relying solely on agent autonomy.

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 2, March-April 2026 ||

DOI:10.15680/IJARETY.2026.1302030

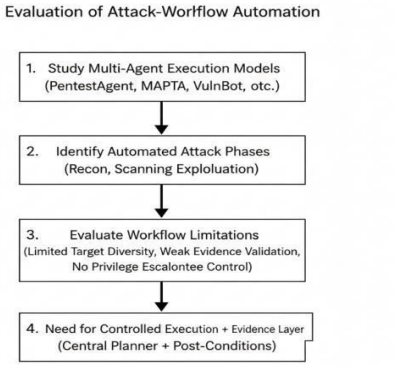


Fig. 3.1 presents the approach taken in evaluating automation of an attack workflow highlighting that an execution proceeds in phases, where domain constraints bite and where evidence-based control of exploitation shows incompleteness.

3.3 Planning-Driven Coordination for Real Exploitation

Offensive frameworks focused on planning, such as Aurora or various unified modeling proposals, create structured decision flows to generate multi-step attack paths derived from system data and tool documentation [3], [16]. These efforts introduce critical methodological components:

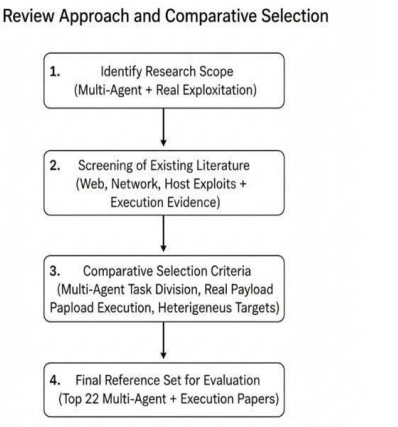
- Attack objectives are defined prior to execution;
- Sequences of tasks adhere to strict dependency constraints; • The feasibility of an attack is assessed before command dispatch.

Despite these structural advantages, planning-based systems rarely deliver actual exploitation or verifiable proof-of-compromise. Attack chains typically terminate at simulation or tool recommendation, failing to capture evidence from executed payloads. Consequently, the methodological synthesis highlights a disparity between robust planning models and agent-driven execution in live environments.

Recognizing this gap points toward a methodological approach that integrates:

- Centralized planning logic;
- Specialized agents for execution;
- Mandatory verification of post-conditions at each stage.

This combination ensures continuity from the planning of attack decisions to the validation of compromise results—an outcome unattained by simulation-driven BAS or decentralized multi-agent strategies.



| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 2, March-April 2026 ||

DOI:10.15680/IJARETY.2026.1302030

Fig. 3.2 depicts a planning-driven coordination model that ties clear attack objectives to execution carried out by specialized agents, and includes post-condition validation to ground real exploitation.

3.4 Summary Synthesis of Methodological Insights

Existing literature identifies two methodological extremes: decentralized multi-agent execution [1]–[5] and centralized planning devoid of live exploitation [3], [16]. Although both approaches advance automation, neither satisfies the requirements for a full-chain operational assessment. Decentralized agents fail to provide coherent oversight for post exploitation and reporting [6], whereas simulation-only planning neglects payload execution and validation entirely. Therefore, a review-based synthesis defines an integrated methodology where:

- Planning dictates agent activities,
- Specialized agents manage action execution, and
- Validation verifies the success of attacks and captures evidential proof.

This tripartite model forms the research foundation for execution-oriented automation capable of replicating realistic red-team activities across host, web, and network surfaces, reflecting true attacker behaviour beyond simple simulations or isolated tasks.

IV. CONCLUSION

Recent inquiries into offensive automation reveal that multi-agent systems have markedly improved the allocation of tasks during reconnaissance, vulnerability identification, and exploit execution. While frameworks such as PentestAgent and MAPTA demonstrate that cooperative agents can reduce the need for human input during initial exploitation phases, their operational scope is often restricted to specific attack surfaces or incomplete workflows [1], [2]. Other extended models, including VulnBot and xOffense, have integrated shared offensive knowledge to improve exploit selection; however, decision-making authority largely remains with individual agents rather than flowing through a unified planning entity [3], [5]. In parallel, environments based on reinforcement learning— like CyberBattleSim and PenGym—have expanded attack adaptability, yet they continue to rely on simulated reward structures that struggle to translate into validated, tool-based exploitation in real-world scenarios [9], [11].

A recurring constraint across these diverse methodologies is the absence of a single, unified orchestration layer capable of governing the entire attack chain—from selecting objectives down to verifying post-conditions. Current implementations tend to either simulate attacks without actual execution [3], [16] or assign tasks to independent agents without guaranteeing evidence-backed outcomes [6]. This review highlights a methodological gap, reinforcing the practical necessity of a centralized planner. Such a component would coordinate specialized agents to carry out live attacks while securing verifiable proof of compromise across host, web, and network environments. Adopting this strategy addresses specific challenges found in decentralized systems, particularly when transitioning between exploitation, privilege escalation, and reporting, where validation must rely on observed system states rather than simulated predictions.

A centralized, execution-focused model offers a more robust foundation for automated red teaming, ensuring that compromises are not merely attempted but empirically proven. By integrating structured planning with task-specific agents and evidence-based validation, full-chain assessments can more accurately mirror the behavior of actual attackers within target infrastructures. This review suggests that extending this model—specifically to cover broader environments, refine adaptive planning, and standardize reporting— represents a critical path toward scalable, reproducible automated red-teaming systems rooted in genuine attack execution instead of abstract simulation.

REFERENCES

[1] F. Shen, Z. Liu, R. Wu, X. Hu, H. Zhu, and K. Chen, "PentestAgent: Incorporating LLM Agents to Automated Penetration Testing," arXiv preprint arXiv:2411.05185, 2024.
[2] R. David and V. Gervais, "MAPTA: Multi-Agent Penetration Testing AI for the Web," arXiv preprint arXiv:2508.20816, 2025.
[3] C. Guo, H. Li, Y. Zhang, and L. Chen, "VulnBot: Autonomous Penetration Testing for a Multi-Agent Collaborative Framework," arXiv preprint arXiv:2501.13411, 2025.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 2, March-April 2026 ||

DOI:10.15680/IJARETY.2026.1302030

[4] L. Zhang, M. He, J. Jiang, Q. Chen, and Y. Du, "xOffense: An AI-Driven Autonomous Penetration Testing Framework with Offensive Knowledge-Enhanced Multi-Agent Systems," arXiv preprint arXiv:2509.13021, 2025.
[5] W. Hu, F. Liu, J. Xu, and P. Yu, "Shell or Nothing: Real-World Benchmarks and Memory Activated Agents for Automated Penetration Testing," arXiv preprint arXiv:2509.09207, 2025.
[6] X. Chen, Z. Li, M. Liu, Z. Zhang, and K. Wang, "AutoRedTeamer: Autonomous Red Teaming with Lifelong Attack Integration," arXiv preprint arXiv:2503.15754, 2025.
[7] RapidPen Research Team, "RapidPen: Fully Automated IP-to-Shell Penetration Testing," arXiv preprint arXiv:2502.16730, 2025.
[8] Y. Duan, M. Yang, J. Li, and F. Yan, "PentestGPT: Evaluating and Harnessing Large Language Models for Pentesting," arXiv preprint arXiv:2308.06782, 2023.
[9] Microsoft Security Research Team, "A Multiagent CyberBattleSim for RL Cyber Operation Agents," arXiv preprint arXiv:2304.11052, 2023.
[10] J. Lin, H. Sun, S. Zhao, and J. Wang, "Multi-Agent Reinforcement Learning in Cybersecurity: From Fundamentals to Applications," arXiv preprint arXiv:2505.19837, 2025.
[11] M. Alkhoulani, A. Althobaiti, and S. Althobaiti, "PenGym: A Reinforcement Learning Training Framework for Pentesting," in Proc. ICISSP, pp. 381–392, 2024.
[12] B. Galvin, S. Mehta, and S. Nambiar, "Autonomous Penetration Testing Using Reinforcement Learning," SSRN Preprint 5208526, 2025.
[13] S. Feng, L. Wang, and T. Zhang, "Raijū: Reinforcement Learning-Guided Post Exploitation Automation," arXiv preprint arXiv:2309.15518, 2023.
[14] N. Benito and A. B. Shaffer, "An Automated Post-Exploitation Model for Cyber Red Teaming," Naval Postgraduate School Research Report, 2021.
[15] Z. Peng, X. Luo, T. Jiang, and L. Huang, "Towards Automated Penetration Testing," arXiv preprint arXiv:2410.17141, 2024.
[16] O. Ramirez, K. Kim, and H. Lee, "A Unified Modelling Framework for Automated Penetration Testing," arXiv preprint arXiv:2502.11588, 2025.
[17] Adversa AI Lab, "Continuous AI Red Teaming for Agentic Systems," Adversa Research Report, 2025.
[18] J. Wang, X. Wu, S. Xu, and F. Chen, "From Sands to Mansions: Simulating Full Attack Chain with LLM-Organized Knowledge," arXiv preprint arXiv:2407.16928, 2024.
[19] C. Artho, J. Hosokawa, and A. Bosu, "Penetration Path Planning in Automated Penetration Testing — Survey," Appl. Sci., vol. 14, no. 18, pp. 8355, 2024.
[20] Microsoft Research, "CyberBattleSim: RL-Based Cybersecurity Simulation," arXiv preprint arXiv:2304.11052, 2023.
[21] T. Wu, X. Zhao, and B. Li, "Graph Mining for Cybersecurity: A Survey," arXiv preprint arXiv:2304.00485, 2023.
[22] O. Udeani, B. Maduka, and N. Nathan, "A Review on the Effectiveness of Red Teaming Exercises in Cybersecurity," World J. Adv. Res. Rev., vol. 18, no. 01, pp. 198–207, 2025.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975 Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com