On the Potential of LLMs for Offensive Security: Benchmarks vs. Operational Reality

Ruben Missotten*†, Vera Rimmer*, Wim Mees†, and Lieven Desmet*
*Distrinet, KU Leuven, Leuven, Belgium
†Cylab, Royal Military Academy, Brussels, Belgium
r.missotten@cylab.be, vera.rimmer@kuleuven.be, w.mees@cylab.be, lieven.desmet@kuleuven.be

***Abstract*—Large Language Models (LLMs), through their strong capabilities in code generation, reasoning, and tool use, have demonstrated promising results in security tasks involving vulnerability discovery and exploitation. However, evaluating their offensive potential in automating penetration testing—a more complex and multi-stage process—remains a critical research challenge. While existing evaluation frameworks effectively demonstrate LLM capabilities in isolated or simplified scenarios, they often do not extend toward the complexity of interconnected attack chains characteristic of real-world adversarial operations. In this analytical study, we examine the challenge of assessing the feasibility of LLM-powered automation across the full adversarial pipeline within realistic environments. We contribute an analysis of current benchmarks and associated environments, and highlight opportunities for methodological enhancements that would strengthen alignment between academic evaluations and operational realities.**

***Index Terms*—Benchmark, Cyber Kill Chain, LLM, MITRE ATT&CK, offensive security, penetration testing, red teaming.**

1. Introduction

The increasing code generation and reasoning capabilities of Large Language Models (LLMs) have led to a rapid adoption of LLMs in cybersecurity. Security research and industry reports show that LLMs are already being leveraged by real-world attackers and even repackaged and sold on underground markets. LLM tools are advertised for phishing, fraud, exploitation and evading detection [1], [2]. By lowering the skill barrier and further enabling automation, these tools increase the offensive potential of AI in the wild.

Building on this concern, researchers and security companies began exploring how offensive capabilities of LLMs can be leveraged in controlled defensive contexts. Most notably, the integration of LLMs into *automated penetration testing* has generated considerable interest and already demonstrated remarkable results: the commercial AI penetration testing system XBOW secured the first place on HackerOne's global leaderboards [3]. LLM-assisted fuzzing has also yielded substantial zero-day vulnerability discoveries [4], [5].

Although significant advancements have been achieved on the both sides of the security arms race, the full extent to which LLMs can autonomously perform or enhance penetration testing remains largely unexplored. Real-world evidence emerges from a few isolated scenarios, while academic research on LLM-assisted penetration testing is limited in both scope and design, and lacks systematization. Overcoming this constrained view and deepening our understanding of the offensive potential of LLMs is a crucial objective toward automating defenses and anticipating emerging threats.

Goal and Contributions. In this paper, we recognize that enterprise penetration testing contexts offer valuable benchmarking potential due to distinctive operational elements which merit greater evaluation focus (e.g., domain-integrated authentication, network segmentation, and multi-host lateral movement). To the best of our knowledge, no systematic analysis has examined the current state of LLM-based penetration testing benchmarks in relation to such real-world operational requirements. The goal of this paper is hence to *analyze existing benchmarks, with particular attention to their alignment to multi-stage attack scenarios spanning the entire Cyber Kill Chain*.

The key contributions are as follows:

- We introduce an *operational perspective on offensive security* that is largely absent from existing literature on penetration testing benchmarks. We argue that enterprise penetration testing is defined by adversarial workflows that set the baseline against which automated solutions must be evaluated (§2). Within this context, we discuss the emerging potential of LLMs for automating offensive security and why current attempts struggle to meet real-world operational demands (§3).
- We analyze *state-of-the-art benchmarking approaches* used to evaluate LLM-based penetration testing in the existing literature, identifying specific gaps between academic benchmarks and enterprise operational requirements (§4).
- We illustrate the identified limitations by systematically mapping six prominent and emerging benchmarks to the MITRE ATT&CK Enterprise Matrix [6], demonstrating partial and uneven coverage of evaluated scenarios across adversarial tactics and techniques (§5). Through this mapping, we identify areas where current benchmarks can expand their *coverage of post-exploitation tactics*, improve *consistency*, and enhance *transparency*.

2. Operational Context

In this section, we establish the conceptual foundation and terminology surrounding the operational realities of

penetration testing and its automation, highlighting primary challenges. We then motivate why adversaries rarely target a single machine, recognizing that *lateral movement across multiple hosts* constitutes a core component tactic.

Cyber Kill Chain. Understanding how real-world adversarial operations unfold is necessary to obtain the required operational insights. The Cyber Kill Chain framework structures cyber attacks along seven stages: *reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives*. Each stage corresponds to a specific set of adversarial Tactics, Techniques, and Procedures (TTPs). The first exploitation of a vulnerability, often referred to as *initial exploitation*, establishes the adversary's foothold within the target network. Typically, this phase is followed by the installation of new malware, establishing a persistent command and control channel. This facilitates lateral movement across multiple hosts within the network, ultimately leading to action on objectives such as the deployment of ransomware or data exfiltration. While the reconnaissance stage may demand significant time investment, particularly in highly targeted attacks, it is upon the transition from initial exploitation to *post-exploitation* that the adversary begins the intensive discovery process: scanning within an unfamiliar network environment. Due to limited resources and other operational constraints, discovery process may be protracted, delaying the adversary's progression toward final objectives.

Terminology and Scope. To clarify the exact scope of offensive security in this paper, we emphasize the distinction between *vulnerability assessment*, *penetration testing*, and *red teaming* based on their complementary approaches. Vulnerability assessment systematically enumerates potential weaknesses across a broad spectrum (emphasizing *breadth*), penetration testing targets specific vulnerabilities to provide empirical validation and impact assessment (*depth*), while red teaming simulates comprehensive adversarial campaigns to evaluate organizational detection and response capabilities across people, processes, and technology (*realism*).

Despite extensive vulnerability assessments by security firms and researchers, poor patch management leaves governments and industries vulnerable. The *Exploit Wednesday* phenomenon—where attackers rapidly reverse engineer Microsoft's *Patch Tuesday* updates to weaponize disclosed flaws—highlights the risk posed by the lag between disclosure and remediation. A Carnegie Mellon study on 75,807 Common Vulnerabilities and Exposures (CVE) IDs found that for vulnerabilities with public exploits, the median time to exploit release is just two days, with 75% appearing within 28 days [7]. This recurring remediation delay underscores the disconnect between vulnerability assessment and effective risk mitigation. Beyond patching delays, system misconfigurations and insecure design practices further compound organizational cyber resilience deficiencies. While highly valuable for addressing the exploitation phase of the Cyber Kill Chain, vulnerability assessment represents *only one component* of achieving cyber resilience.

Vulnerability assessment and penetration testing may focus on specific domains—e.g., web applications, mobile platforms, cloud infrastructure, or APIs—while both penetration testing and red teaming can address entire attack surfaces. Red teaming emphasizes stealth operations and Advanced Persistent Threat (APT) simulation, providing maximum value for organizations with mature security controls and incident response procedures. The appropriate focus finally depends on target audience and testing objectives, though *enterprise penetration testing often offers a practical middle ground that balances depth and feasibility.*

Realism. Realistic security evaluations requires adhering to black-box principles, where no prior environmental knowledge is assumed. However, such simulations encounter numerous practical challenges, more so in certain specific breach scenarios—e.g., with user-enabled vectors such as phishing that prove difficult to reliably simulate. The complexity increases when considering offensive campaigns spanning several months, complicating reproducibility and evaluation consistency. When evaluating automated tools (such as AI agents) under these conditions, the inherent stochastic properties of both the agent's model and the testing environment lead to highly divergent outcomes. This progression of complications directly impacts the *appropriate extent of realism to pursue*, inevitably leading to pragmatic compromises in both evaluation and agent training contexts.

Environment Design. Understanding the target audience for a security evaluation is instrumental for designing an appropriately dynamic environment for tests. Microsoft Windows remains central to enterprise user management through Active Directory, with an increasing array of services deployed on the Azure cloud platform. Hence, with the bulk of end hosts using Windows [8], omitting Windows is typically unrealistic. Expanding attack surfaces—including mobile devices, IoT integrations, and Bring-Your-Own-Device (BYOD) policies—add further complexity. Embedded systems with legacy software and sectors focused on Operational Technology (OT) underscore the value of including technological diversity. Depending on the use case, a targeted and restricted technology scope may enable deep, comprehensive evaluation, but may limit the assessment of an offensive agent's broader capabilities. We argue that an ultimate effective test environment should comprise a *network of interdependent systems*, facilitating accurate assessment of lateral movement and network pivoting behaviors—capabilities that single-host environments cannot sufficiently capture.

Distribution of Adversarial Effort. Academic research has made significant contributions with regard to exploitation of unknown vulnerabilities (zero-days) [9]. Meanwhile, industry observations reveal that cyber intrusions frequently leverage known (n-day) vulnerabilities, stolen credentials, and phishing for initial access [10]–[12]. Empirical evidence further underscores the disparity between academic research and the operational realities observed in the field. In 2023, Mandiant reported an average time-to-exploit[1] of five days [13], indicating foothold acquisition can occur rapidly, as it entails scanning numerous hosts and selectively targeting vulnerable ones. In contrast, the global median dwell time[2]

1. Mean time required to exploit a vulnerability relative to patch release.
2. Number of days an attacker remains undetected within an environment.

was 11 days in 2024 (down from 205 days in 2014) [11], reflecting sustained, multi-day post-exploitation operations that mostly precede attainment of objectives. A substantial amount of adversaries fail to achieve their objectives prior to detection, so that the actual time they would spend—once inside a network—is several times greater. According to a 2016 study conducted by SmokeScreen, adversaries allocate approximately 80% of their operational time to lateral movement [14]. Taken together, these statistics strongly indicate that, irrespective of entry vector (through zero-days or known vulnerabilities), *post-exploitation dominates attacker labor and elapsed time*, whereas enterprise penetration tests and red team engagements—constrained by contractual time limits—rarely replicate such extended lateral operations. Furthermore, in practice, attackers prefer to adopt living-off-the-land techniques [15], leveraging legitimate, native binaries present on the victim's system that are misconfigured or poorly protected to perform malicious actions. They can evade detection by operating entirely in memory, hence leaving minimal traces on disk. This approach showcases adversaries' advanced proficiency in repurposing existing utilities rather than developing custom frameworks, achieving compromises without any malware deployment.

Takeaway 1. Insights from industry present valuable opportunities for academia to realign offensive evaluation methods with realistic adversarial behaviors and operational realities. Particularly in the context of automated enterprise penetration testing, the presented factors motivate a greater research emphasis on *accurately representing and automating the post-exploitation segment of the Cyber Kill Chain*.

3. Automation and LLMs

A lot of progress has been made in automating penetration testing processes. Of all machine learning paradigms and applications, LLMs seem most promising due to their ability to interpret human-readable text, whether previously seen or unseen. Their inherent non-deterministic properties, even under deterministic settings (i.e., temperature equal to zero [16]), serve as a significant enabler for exploration. Several factors support the suitability of LLMs. First, remote system interaction primarily relies on Command-Line Interfaces (CLIs), which administrators and developers originally designed to be human-readable for effective troubleshooting—a principle that extends naturally to remote API interactions. Second, computer systems exhibit inherent variability in their responses to commands and code execution, stemming from the dynamic, concurrent execution of interdependent threads and processes. This stochastic behavior aligns well with LLMs' probabilistic nature. Moreover, emerging multimodal learning enhances practical applications by supporting Graphical User Interfaces (GUIs), enabling these systems to more closely approximate human penetration tester workflows.

Groundwork on LLMs for penetration testing predominantly employs foundation models, which demonstrate considerable domain knowledge across the Cyber Kill Chain, particularly regarding known vulnerabilities and established tools supported by extensive tutorial resources. Models are occasionally fine-tuned on custom datasets [17] or expanded via Retrieval-Augmented Generation (RAG) with Capture the Flag (CTF) writeups [18], CVE reports [19] or TTP-related information [20]. Recent developments in agentic methodologies and reasoning models have enhanced penetration testing automation by reducing task complexity [19], though we argue that they necessitate increased computational resources and a robust framework for hierarchical task orchestration.

Empirical studies reveal that LLM-driven penetration testing presents unique challenges due to context window constraints, recency bias, hallucination tendencies, and maintaining holistic reasoning across large, dynamic environments [21]. These factors can lead to suboptimal command execution, creating opportunities for improving LLM consistency in reaching target states [22], while also risking to raise detection rates in red teaming context. Complex scenarios illustrate these challenges: when an agent discovers credentials using a tool such as Mimikatz, it must both retain this information and understand its strategic value many steps further along the attack chain. This can result in exceeding token limits as context accumulates or in excessive vectorization leading to critical details being lost. Leveraging explainable AI, research begins to address these areas for improvement [23], contributing to a deeper understanding of LLM behavior in complex operational contexts.

XBOW, a commercial LLM–based penetration testing framework, achieved the top position on HackerOne's global leaderboards [3]. An increasing number of models exhibit enhanced reasoning by engaging in prolonged deliberation prior to generating scripts or commands. However, they observe this approach proving counterproductive in penetration testing contexts, which requires rapid, iterative exploration, discovery, and information gathering [24]. While they demonstrate performance using both proprietary benchmarks and the renowned bug bounty platform, the evaluation focuses on specific domains—web, cloud, and mobile vulnerability exploitation—without post-exploitation activities.

Takeaway 2. We recognize an insufficient exploration of offensive LLM agents within a *realistic interconnected network of hosts characterized by causal dependencies between actions*. Such a setup would better reflect the complexity of enterprise environments, building upon current benchmarks that address single-machine targets and isolated vulnerable services. A similar shift from perimeter-focused security, as seen in the nineties, could apply here. Yet, despite long-standing defense-in-depth adoption, comparable layered strategies for automated offensive tools remain noticeably absent. While current research demonstrates specific capabilities of LLMs in penetration testing scenarios [25], further progress demands more representative evaluation approaches. An essential factor is the *quality of benchmarks*—the focus of the remainder of this paper—as it facilitates the proper measurement, validation, and comparison of advances.

4. Current State of Benchmarks

This section analyzes existing LLM-driven penetration testing benchmarks. First we distinguish *knowledge-based*

benchmarks—that assess static knowledge retrieval—from *task-based benchmarks*—that test actual interactive task execution by agents. Then we examine the current state of *multi-host benchmarks*: task-based benchmarks that aim to emulate complex interconnected enterprise environments.

4.1. LLM Benchmark Types

Knowledge-Based. Conventionally, general-purpose LLMs are assessed for multi-domain knowledge through the use of question-answering datasets, wherein questions are formulated by domain experts. They often necessitate extended reasoning by the model, with results commonly reported on leaderboards. Such *knowledge-based* benchmarks include the Massive Multitask Language Understanding Contamination-Free (MMLU-CF) [26] and Humanity's Last Exam [27]. They use private test sets—kept inaccessible during model training or tuning and reserved for final evaluation and scoring—alongside public or semi-public validation sets for peer benchmarking and community assessments.

Domain-specific cybersecurity benchmarks also exist, such as SecQA [28], which represents a concise, security-focused multiple-choice benchmark derived from a single authoritative textbook. It incorporates two levels of difficulty and human validation to achieve targeted, pedagogically structured coverage. CyberMetric [29] constitutes a more expansive, retrieval-grounded multiple-choice benchmark constructed through RAG over standards, RFCs, academic papers, and books, with expert review to ensure broad coverage and traceable sourcing. Both are openly accessible for reproducible evaluation, though SecQA's reliance on a single source may introduce coverage bias. The Weapons of Mass Destruction Proxy (WMDP) knowledge-based benchmark provides a valuable public benchmark for cybersecurity hazardous knowledge evaluation, covering the full offensive attack lifecycle from reconnaissance through post-exploitation [30]. Regardless, any public availability inherently heightens contamination risk through memorization.

Task-Based. Current evaluations of LLM-driven agents tend to emphasize static, *knowledge-based* assessments, including those recently introduced by prominent industry actors [31], [32]. However, enterprise networks operate as dynamic systems and require interactive engagement due to their stochastic nature. Consequently, *task-based* benchmarks are emerging, where agents must interact with environments whose behavior is contingent on the agent's prior actions. We observe that many existing task-based benchmarks represent predominantly gamified, CTF-style environments [19], [33]–[39]. Others—while not strictly CTF-oriented—still focus on single-machine targets or isolated vulnerable services [40]–[42]. While these setups effectively illustrate core exploitation techniques, they preclude realistic lateral movement across multiple hosts—a fundamental aspect of post-exploitation.

CTF challenges, from popular platforms such as HackTheBox [43] and Vulnhub [44], are deliberately engineered to be overtly vulnerable and include explicit hints—features rarely encountered in operational environments. On top of this oversimplification, we notice that the CTF challenges are selected according to opaque, inconsistent criteria and hence, may be subject to selective snooping [45]. Finally, the incorporation of public CTF writeups into model training data, raises contamination concerns [38] and can inadvertently lead to test snooping [45], potentially inflating reported performance on publicly accessible benchmarks in prior work [17], [46]. Together, these common pitfalls put under question the generalization of findings yielded in CTF-driven benchmarks toward realistic environments.

Frameworks like OCCULT appear to bridge knowledge-based and task-based approaches through a tripartite evaluation methodology for assessing offensive LLM agents [47]. This includes scenario-driven multiple-choice questions, tasks involving synthetic Windows Active Directory data analysis, and network simulations for multi-step attack sequences. To mitigate contamination concerns, the framework reportedly employs dynamic variable generation and procedurally generated network topologies. While this simulation-based approach may provide computational efficiency and standardized metrics, it differs from emulation by abstracting low-level kernel interactions and limiting testing to predefined action spaces.

4.2. Toward Multi-Host Benchmarks

Research emphasizes the necessity for realistic, interconnected multi-host scenarios, noting that existing benchmarks do not accurately reflect real-world conditions and highlighting the inadequate representation of certain aspects of the Cyber Kill Chain [48]. Furthermore, research states that benchmarks are misaligned with understanding real-world impact because they focus solely on measuring model capabilities rather than conducting comprehensive risk assessments [49]. Happe and Cito present a proof of concept examining whether LLMs are capable of compromising an enterprise network by provisioning a tailored environment—Game of Active Directory (GoAD) [50]—based on Microsoft Windows Active Directory [51]. Despite the topology being a simplified abstraction of an enterprise network and the environment itself being intentionally vulnerable, the authors observe that LLMs frequently pursue meaningless strategies and lose track of context. By emphasizing metrics such as operational cost and token consumption, the work underscores the necessity of reproducibility.

Recent advancements in emulated, multi-host, dynamic benchmarks demonstrate that, even within relatively simple environments, current models and automated penetration-testing frameworks commonly struggle to conduct multi-host attacks [22], [51]. The authors of AutoAttacker evaluate an autonomous penetration testing agent using an interconnected network of multiple hosts [52]—demonstrating progress toward more realistic benchmarks. They note that their victim environment was intentionally configured with certain weaknesses to facilitate successful attacks, reflecting a controlled laboratory setup with a limited subset of tasks due to practical research constraints. Singer et al. introduced MHBench [22], a comprehensive benchmark comprising ten

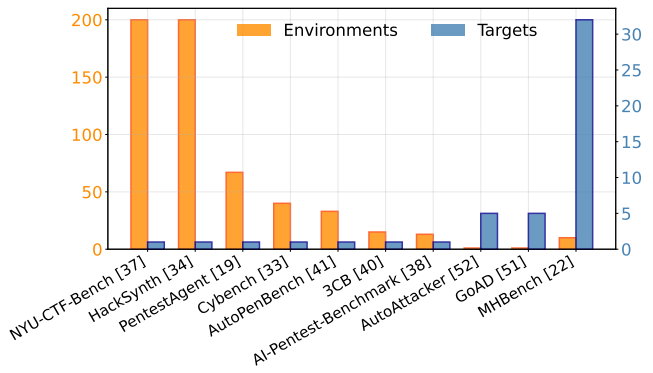

Figure 1. **Environments** denote the approximated total number of scenarios; **targets** denote the average number of target machines per scenario.

different environments with 25–50 hosts each. The evaluation incorporates authentic breach incidents and enterprise architectures, providing valuable insights despite focusing on a limited amount of vulnerabilities and network topologies. LLMs face notable challenges with shell-based multi-host attacks, achieving minimal success. The authors propose an abstraction layer separating attack planning from execution which measurably enhances LLM effectiveness.

As discussed in Section 2, we assume that realism strongly correlates with the extent of lateral movement possible between machines. Hence, to conclude this section, we categorize emerging task-based benchmarks based on the included targets and environments, as shown in Figure 1. Here, *targets* denote the average number of machines or hosts that an agent can access laterally within a given scenario. We define an *environment* as a single scenario—typically a challenge, task, or a set of subtasks. This is often implemented as a container or, ideally, a virtual machine enabling broader access to system resources and less restricted to Linux-only deployments. Including a large set of targets will possibly enable a more thorough evaluation of post-exploitation phases, including the assessment of extended attack chains. Testing with a larger number of environments intuitively implies greater scenario diversity and broader coverage of tactics and techniques from the MITRE ATT&CK framework. While this may be expected, the relationship is not always straightforward (cf. Section 5). Many targets and environments can be repetitive and still correspond to isolated tasks, restricting the number of steps an agent can perform within a single environment and limiting realism in relation to the Cyber Kill Chain.

Takeaway 3. Our analysis reveals a clear *heterogeneity in evaluation environments and target configurations* across benchmarks, reflecting divergent priorities. Enhanced scenario diversity creates trade-offs with target density within individual environments due to practical constraints. While maximizing both counts is highly desirable for a comprehensive Cyber Kill Chain representation, we argue that *target density remains paramount*: a limited number of targets fundamentally undermines post-exploitation assessment.

5. Systematic Assessment

This section reports on our analysis of *coverage for current task-based penetration testing benchmarks* with respect to the MITRE ATT&CK Enterprise Matrix [6]. We situate emerging benchmarks within the entire landscape (1) to evaluate completeness of each benchmark across the full process of an attack, and (2) to identify systematic coverage gaps that may inform the design of more comprehensive future benchmarks. To the best of our knowledge, this perspective on systematically assessing and comparing benchmarks is novel: it provides a complementary view to prior work [48], which focuses on extensively characterizing each individual benchmark without explicitly reflecting completeness.

In order to convey the general scope and possible gaps of current benchmarks, we selected the benchmarks from Figure 1 considered most representative for assessing coverage (i.e., considerably advanced and well-documented at the time of writing). The AI-Pentest-Benchmark [38], NYU-CTF-Bench [37], Catastrophic Cyber Capabilities Benchmark (3CB) [40], GoAD used by Happe and Cito [51], and the benchmark put forward by AutoAttacker [52] explicitly specified MITRE ATT&CK techniques, tactics, or related concepts that could be applied to their environment—articulating the evaluative focus intended by the authors. While MHBench [22] neither disclosed this information nor released its source code at the time of writing, our assessment shows that MHBench currently represents one of the more realistic benchmarks. We performed an independent mapping of MHBench based on environment descriptions presented in the paper. Extracted data and analysis details are accessible online[3], and here we synthesize the main results.

As shown in Figure 2, post-exploitation is underweighted relative to its operational importance, with sparse attention to lateral movement despite its centrality to sustained, multi-host compromise. AI-Pentest-Benchmark, though broad in host-level activities, concentrates heavily on pre-exploitation reconnaissance and early steps, leading to virtually no treatment of persistence and lateral movement that constrains assessment of agents' sustained foothold and progression capabilities. In NYU-CTF-Bench, we argue that the measurable persistence coverage within a predominantly single-host CTF corpus likely reflects categorical mapping of challenge mechanics rather than enterprise-grade persistence and, notably, omits lateral movement evaluation. Their techniques largely focus on defense evasion, requiring greater maturity for implementation. 3CB demonstrates broad coverage; however, as shown in Figure 1, it features approximately one target per scenario. While this approach thoroughly evaluates specific techniques, it tends to emphasize discrete, tool-driven steps over interdependent, longitudinal attack chains. Despite modest scale, the GoAD environment includes a complex Windows Active Directory structure, supporting focused evaluation of post-exploitation tactics. In contrast, the AutoAttacker benchmark reports persistence and other post-breach tactics as relatively isolated tasks, producing persis-

3. https://cylab.be/waiti2025

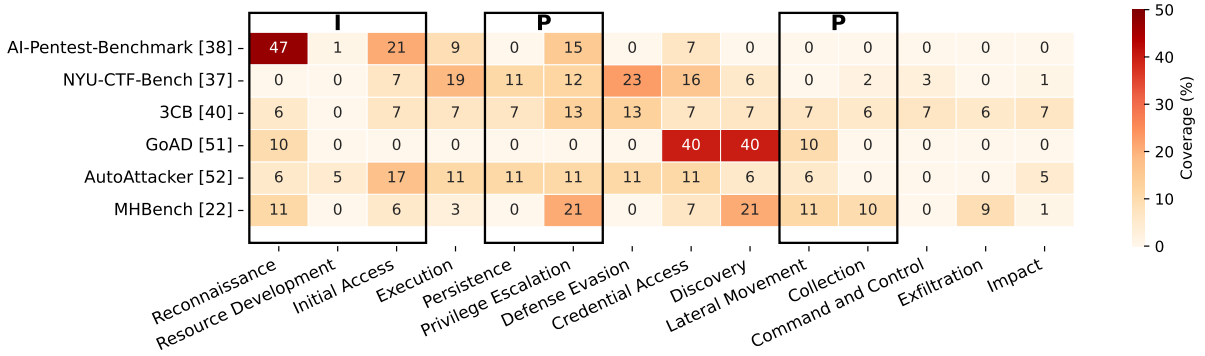

Figure 2. Coverage of **MITRE ATT&CK Enterprise tactics** across the selection of **benchmarks**. The percentages show the proportion of tasks within a benchmark that represents a given tactic. The borders represent Initial Compromise (**I**) and Post-Compromise (**P**) stages.

tence without robust pathways for movement and discovery-driven expansion. MHBench appears strongest conceptually, introducing multi-host networks and end-to-end sequences that better reflect post-exploitation realities. However, the absence of a Windows ecosystem limits scenario diversity.

As a step toward systematic comparison, we divide MITRE ATT&CK tactics into temporal phases based on access requirements. We classify tactics up to and including initial exploitation as *Initial Compromise* ('I' on Figure 2: reconnaissance, resource development, and initial access) and those strictly post-exploitation as *Post-Compromise* ('P' on Figure 2: persistence, privilege escalation, lateral movement, and collection), as each presupposes an existing foothold. The remaining tactics typically commence at or shortly after compromise and can recur throughout an intrusion, being not contingent on extended dwell time. Table 1 groups the percentages of Initial and Post-Compromise coverage.

TABLE 1. DISTRIBUTION OF INITIAL AND POST-COMPROMISE TACTICS FOR EACH BENCHMARK. PERCENTAGES INDICATE THE RESPECTIVE APPROXIMATED PROPORTION FOR EACH TACTIC CATEGORY.

Benchmark	Initial Compromise (I): 3 tactics (%)	Post-Compromise (P): 4 tactics (%)
AI-Pentest-Benchmark [38]	69	15
NYU-CTF-Bench [37]	7	25
3CB [40]	13	33
GoAD [51]	10	10
AutoAttacker [52]	28	28
MHBench [22]	17	42

Takeaway 4. Table 1 reveals that *Post-Compromise tactics are underrepresented relative to their operational significance*. As discussed in Section 2, adversaries spend roughly 80% of their time on lateral movement—an imbalance not reflected in current benchmarks. Beyond asserting broad coverage, advanced evaluations should test *coherent attack chains rather than isolated tactics*—e.g., any measured persistence should be accompanied by realistic opportunities for lateral movement and subsequent objectives.

Takeaway 5. The inconsistencies across Figures 1, 2 and Table 1 signal *inadequate standardization*. Environment and task selection appear largely discretionary, producing selectively curated scenarios with uneven coverage. The objective here is not to achieve a uniform coverage but to ensure stronger coverage of key tactics, such as the Post-Compromise ones, where attackers devote most effort. Crucially, even adequate coverage does not guarantee benchmark effectiveness, as individual tasks may lack complexity or realism. Such mappings do not capture task interdependencies, regardless of whether complex scenarios are included. Nevertheless, we argue that establishing a form of mapping is the minimum prerequisite for comparing benchmarks.

Takeaway 6. Lastly, we observe that *publicly released benchmarks often omit detailed environmental descriptions*, impeding comparison and external validation. Improving transparency could involve the adoption of a *model card* or *system card*, analogous to recommendations for open-sourcing an LLM [53]. We recommend addressing this prior to undertaking a study on a novel benchmark.

6. Conclusion

LLM-driven offensive security benchmarks have laid crucial groundwork on feasibility of automated penetration testing. While current benchmarks offer valuable baselines, our analysis reveals substantial heterogeneity in their design and a predominant focus on static evaluations, isolated tactics, or simplified single-host setups—conditions that diverge from the operational realities of compromise. To support the next phase of progress, we identify three areas of research focus:

1) stronger emphasis on post-exploitation benchmarks;
2) systematic evaluation of multi-stage attack chains rather than disjoint tasks or individual challenges;
3) standardized, transparent reporting of benchmark coverage according to established frameworks.

Establishing these foundations is a prerequisite for reproducible, operationally relevant benchmarking and research on advancing offensive LLM agents to navigate the full complexity of real-world adversarial operations.

Acknowledgment

This research is partially funded by the Research Fund KU Leuven, and the Cybersecurity Research Program Flanders.

References

[1] M. F. Mohamed Firdhous, W. Elbreiki, I. Abdullahi, B. Sudantha, and R. Budiarto, "WormGPT: A large language model chatbot for criminals," in *2023 24th International Arab Conference on Information Technology (ACIT)*, 2023, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ACIT58888.2023.10453752

[2] Z. Lin, J. Cui, X. Liao, and X. Wang, "Malla: Demystifying real-world large language model integrated malicious services," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 4693–4710. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/lin-zilong

[3] N. Waisman. XBOW - XBOW on HackerOne: What's next. Published: August 18, 2025. [Online]. Available: https://xbow.com/blog/xbow-on-hackerone-whats-next

[4] R. Meng, M. Mirchev, M. Böhme, and A. Roychoudhury, "Large language model guided protocol fuzzing," in *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*, 2024.

[5] C. S. Xia, M. Paltenghi, J. Le Tian, M. Pradel, and L. Zhang, "Fuzz4All: Universal fuzzing with large language models," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ser. ICSE '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3597503.3639121

[6] MITRE ATT&CK®. [Online]. Available: https://attack.mitre.org/

[7] A. D. Householder, J. Chrabaszcz, T. Novelly, D. Warren, and J. M. Spring, "Historical analysis of exploit availability timelines," in *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*. USENIX Association, Aug. 2020. [Online]. Available: https://www.usenix.org/conference/cset20/presentation/householder

[8] Desktop operating system market share worldwide. StatCounter Global Stats. [Online]. Available: https://gs.statcounter.com/os-market-share/desktop/worldwide/

[9] Y. Zhu, A. Kellermann, A. Gupta, P. Li, R. Fang, R. Bindu, and D. Kang, "Teams of LLM agents can exploit zero-day vulnerabilities," 2025. [Online]. Available: https://arxiv.org/abs/2406.01637

[10] "IBM X-Force 2025 threat intelligence index," IBM Security, published: June, 2025. [Online]. Available: https://www.ibm.com/reports/threat-intelligence

[11] "M-Trends 2025 report," Google Cloud Security, published: April 23, 2025. [Online]. Available: https://cloud.google.com/blog/topics/threat-intelligence/m-trends-2025/

[12] A. Happe and J. Cito, "Understanding hackers' work: An empirical study of offensive security practitioners," in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2023. New York, NY, USA: Association for Computing Machinery, 2023, p. 1669–1680. [Online]. Available: https://doi.org/10.1145/3611643.3613900

[13] C. Charrier and R. Weiner. How low can you go? An analysis of 2023 time-to-exploit trends. Google Cloud Blog. Published: October 15, 2024. [Online]. Available: https://cloud.google.com/blog/topics/threat-intelligence/time-to-exploit-trends-2023

[14] S. Kumbhar, "Top lateral movement techniques: The red team edition," Smokescreen Technologies, Tech. Rep., 2016, white paper originally available at smokescreen.io, URL no longer accessible.

[15] F. Barr-Smith, X. Ugarte-Pedrero, M. Graziano, R. Spolaor, and I. Martinovic, "Survivalism: Systematic analysis of Windows malware living-off-the-land," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1557–1574. [Online]. Available: https://doi.org/10.1109/SP40001.2021.00047

[16] B. Atil, S. Aykent, A. Chittams, L. Fu, R. J. Passonneau, E. Radcliffe, G. R. Rajagopal, A. Sloan, T. Tudrej, F. Ture, Z. Wu, L. Xu, and B. Baldwin, "Non-determinism of "deterministic" LLM settings," 2025. [Online]. Available: https://arxiv.org/abs/2408.04667

[17] P. D. Luong, L. T. G. Bao, N. V. K. Tam, D. H. N. Khoa, N. H. Quyen, V.-H. Pham, and P. T. Duy, "xOffense: An AI-driven autonomous penetration testing framework with offensive knowledge-enhanced LLMs and multi agent systems," 2025. [Online]. Available: https://arxiv.org/abs/2509.13021

[18] M. Shao, H. Xi, N. Rani, M. Udeshi, V. S. C. Putrevu, K. Milner, B. Dolan-Gavitt, S. K. Shukla, P. Krishnamurthy, F. Khorrami, R. Karri, and M. Shafique, "CRAKEN: Cybersecurity LLM agent with knowledge-based execution," 2025. [Online]. Available: https://arxiv.org/abs/2505.17107

[19] X. Shen, L. Wang, Z. Li, Y. Chen, W. Zhao, D. Sun, J. Wang, and W. Ruan, "PentestAgent: Incorporating LLM agents to automated penetration testing," in *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 375–391. [Online]. Available: https://doi.org/10.1145/3708821.3733882

[20] R. Fayyazi, R. Taghdimi, and S. J. Yang, "Advancing TTP analysis: Harnessing the power of large language models with retrieval augmented generation," in *2024 Annual Computer Security Applications Conference Workshops (ACSAC Workshops)*, 2024, pp. 255–261. [Online]. Available: https://doi.org/10.1109/ACSACW65225.2024.00036

[21] A. Happe and J. Cito, "On the surprising efficacy of LLMs for penetration-testing," 2025. [Online]. Available: https://arxiv.org/abs/2507.00829

[22] B. Singer, K. Lucas, L. Adiga, M. Jain, L. Bauer, and V. Sekar, "On the feasibility of using LLMs to autonomously execute multi-host network attacks," 2025. [Online]. Available: https://arxiv.org/abs/2501.16466

[23] T. M. Ghazal, J. I. Janjua, W. Abushiba, M. Ahmad, A. Ihsan, and N. A. Al-Dmour, "Cybersecurity revolution via large language models and explainable AI," in *2024 17th International Conference on Security of Information and Networks (SIN)*, 2024, pp. 1–6. [Online]. Available: https://doi.org/10.1109/SIN63213.2024.10871324

[24] O. de Moor and A. Ziegler. XBOW - XBOW unleashes GPT-5's hidden hacking power, doubling performance. Published: August 15, 2025. [Online]. Available: https://xbow.com/blog/gpt-5

[25] H. Xu, S. Wang, N. Li, K. Wang, Y. Zhao, K. Chen, T. Yu, Y. Liu, and H. Wang, "Large language models for cyber security: A systematic literature review," *ACM Trans. Softw. Eng. Methodol.*, Sep. 2025, just Accepted. [Online]. Available: https://doi.org/10.1145/3769676

[26] Q. Zhao, Y. Huang, T. Lv, L. Cui, Q. Sun, S. Mao, X. Zhang, Y. Xin, Q. Yin, S. Li, and F. Wei, "MMLU-CF: A contamination-free multi-task language understanding benchmark," 2024. [Online]. Available: https://arxiv.org/abs/2412.15194

[27] Phan et al., "Humanity's last exam," 2025. [Online]. Available: https://arxiv.org/abs/2501.14249

[28] Z. Liu, "SecQA: A concise question-answering dataset for evaluating large language models in computer security," 2023. [Online]. Available: https://arxiv.org/abs/2312.15838

[29] N. Tihanyi, M. A. Ferrag, R. Jain, T. Bisztray, and M. Debbah, "CyberMetric: A benchmark dataset based on retrieval-augmented generation for evaluating LLMs in cybersecurity knowledge," in *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2024, pp. 296–302. [Online]. Available: https://doi.org/10.1109/CSR61664.2024.10679494

[30] N. Li, A. Pan, A. Gopal, S. Yue, D. Berrios, A. Gatti, J. D. Li, A.-K. Dombrowski, S. Goel, G. Mukobi, N. Helm-Burger, R. Lababidi, L. Justen, A. B. Liu, M. Chen, I. Barrass, O. Zhang, X. Zhu, R. Tamirisa, B. Bharathi, A. Herbert-Voss, C. B. Breuer, A. Zou, M. Mazeika, Z. Wang, P. Oswal, W. Lin, A. A. Hunt, J. Tienken-Harder, K. Y. Shih, K. Talley, J. Guan, I. Steneker, D. Campbell, B. Jokubaitis, S. Basart, S. Fitz, P. Kumaraguru, K. K. Karmakar, U. Tupakula, V. Varadharajan, Y. Shoshitaishvili, J. Ba, K. M. Esvelt, A. Wang, and D. Hendrycks, "The WMDP benchmark: Measuring and reducing malicious use with unlearning," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, 21–27 Jul 2024, pp. 28 525–28 550. [Online]. Available: https://proceedings.mlr.press/v235/li24bc.html

[31] Z. Liu, J. Shi, and J. F. Buford, "Cyberbench: A multi-task benchmark for evaluating large language models in cybersecurity," AAAI-24 Workshop on Artificial Intelligence for Cyber Security (AICS), 2024.

[32] M. Levi, Y. Allouche, D. Ohayon, and A. Puzanov, "CyberPal.AI: Empowering LLMs with expert-driven cybersecurity instructions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 23, pp. 24 402–24 412, Apr. 2025. [Online]. Available: https://doi.org/10.1609/aaai.v39i23.34618

[33] A. K. Zhang, N. Perry, R. Dulepet, J. Ji, C. Menders, J. W. Lin, E. Jones, G. Hussein, S. Liu, D. J. Jasper, P. Peetathawatchai, A. Glenn, V. Sivashankar, D. Zamoshchin, L. Glikbarg, D. Askaryar, H. Yang, A. Zhang, R. Alluri, N. Tran, R. Sangpisit, K. O. Oseleononmen, D. Boneh, D. E. Ho, and P. Liang, "Cybench: A framework for evaluating cybersecurity capabilities and risks of language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=tc90LV0yRL

[34] L. Muzsai, D. Imolai, and A. Lukács, "HackSynth: LLM agent and evaluation framework for autonomous penetration testing," 2024. [Online]. Available: https://arxiv.org/abs/2412.01778

[35] G. Deng, Y. Liu, V. Mayoral-Vilches, P. Liu, Y. Li, Y. Xu, T. Zhang, Y. Liu, M. Pinzger, and S. Rass, "PentestGPT: Evaluating and harnessing large language models for automated penetration testing," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 847–864. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/deng

[36] M. Kobayashi, M. Fuchi, A. Zanashir, T. Yoneda, and T. Takagi, "Construction and evaluation of LLM-based agents for semi-autonomous penetration testing," 2025. [Online]. Available: https://arxiv.org/abs/2502.15506

[37] M. Shao, S. Jancheska, M. Udeshi, B. Dolan-Gavitt, H. Xi, K. Milner, B. Chen, M. Yin, S. Garg, P. Krishnamurthy, F. Khorrami, R. Karri, and M. Shafique, "NYU CTF Bench: A scalable open-source benchmark dataset for evaluating LLMs in offensive security," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 57 472–57 498. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/69d97a6493fbf016fff0a751f253ad18-Paper-Datasets_and_Benchmarks_Track.pdf

[38] I. Isozaki, M. Shrestha, R. Console, and E. Kim, "Towards automated penetration testing: Introducing LLM benchmark, analysis, and improvements," in *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP Adjunct '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 404–419. [Online]. Available: https://doi.org/10.1145/3708319.3733804

[39] M. Rodriguez, R. A. Popa, F. Flynn, L. Liang, A. Dafoe, and A. Wang, "A framework for evaluating emerging cyberattack capabilities of AI," 2025. [Online]. Available: https://arxiv.org/abs/2503.11917

[40] A. Anurin, J. Ng, K. Schaffer, J. Schreiber, and E. Kran, "Catastrophic cyber capabilities benchmark (3CB): Robustly evaluating LLM agent cyber offense capabilities," 2024. [Online]. Available: https://arxiv.org/abs/2410.09114

[41] L. Gioacchini, M. Mellia, I. Drago, A. Delsanto, G. Siracusano, and R. Bifulco, "AutoPenBench: Benchmarking generative agents for penetration testing," 2024. [Online]. Available: https://arxiv.org/abs/2410.03225

[42] S. Wan, C. Nikolaidis, D. Song, D. Molnar, J. Crnkovich, J. Grace, M. Bhatt, S. Chennabasappa, S. Whitman, S. Ding, V. Ionescu, Y. Li, and J. Saxe, "CYBERSECEVAL 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models," 2024. [Online]. Available: https://arxiv.org/abs/2408.01605

[43] Hack The Box: The #1 cybersecurity performance center. [Online]. Available: https://www.hackthebox.com

[44] Vulnerable by design $\sim$ VulnHub. [Online]. Available: https://www.vulnhub.com/

[45] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 3971–3988. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/arp

[46] R. Turtayev, A. Petrov, D. Volkov, and D. Volk, "Hacking CTFs with plain agents," 2024. [Online]. Available: https://arxiv.org/abs/2412.02776

[47] M. Kouremetis, M. Dotter, A. Byrne, D. Martin, E. Michalak, G. Russo, M. Threet, and G. Zarrella, "OCCULT: Evaluating large language models for offensive cyber operation capabilities," 2025. [Online]. Available: https://arxiv.org/abs/2502.15797

[48] A. Happe and J. Cito, "Benchmarking practices in LLM-driven offensive security: Testbeds, metrics, and experiment design," 2025. [Online]. Available: https://arxiv.org/abs/2504.10112

[49] K. Lukošiūtė and A. Swanda, "LLM cyber evaluations don't capture real-world risk," 2025. [Online]. Available: https://arxiv.org/abs/2502.00072

[50] Game Of Active Directory. [Online]. Available: https://orange-cyberdefense.github.io/GOAD/

[51] A. Happe and J. Cito, "Can LLMs hack enterprise networks? Autonomous assumed breach penetration-testing Active Directory networks," *ACM Trans. Softw. Eng. Methodol.*, Sep. 2025. [Online]. Available: https://doi.org/10.1145/3766895

[52] J. Xu, J. W. Stokes, G. McDonald, X. Bai, D. Marshall, S. Wang, A. Swaminathan, and Z. Li, "AutoAttacker: A large language model guided system to implement automatic cyber-attacks," 2024. [Online]. Available: https://arxiv.org/abs/2403.01038

[53] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. III, and K. Crawford, "Datasheets for datasets," *Commun. ACM*, vol. 64, no. 12, p. 86–92, Nov. 2021. [Online]. Available: https://doi.org/10.1145/3458723