Cybersecurity AI: Evaluating Agentic Cybersecurity in Attack/Defense CTFs

Francesco Balassone[1,2], Víctor Mayoral-Vilches[1], Stefan Rass[3], Martin Pinzger[4], Gaetano Perrone[2], Simon Pietro Romano[2], Peter Schartner[4]

[1]Alias Robotics
[2]Università degli Studi di Napoli Federico II
[3]Johannes Kepler University Linz
[4]Alpen-Adria-Universität Klagenfurt
research@aliasrobotics.com

Abstract

We empirically evaluate whether AI systems are more effective at attacking or defending in cybersecurity. Using CAI (Cybersecurity AI)'s parallel execution framework, we deployed autonomous agents in 23 Attack/Defense CTF battlegrounds. Statistical analysis reveals defensive agents achieve 54.3% unconstrained patching success versus 28.3% offensive initial access (p=0.0193), but this advantage disappears under operational constraints: when defense requires maintaining availability (23.9%) and preventing all intrusions (15.2%), no significant difference exists (p>0.05). Exploratory taxonomy analysis suggests potential patterns in vulnerability exploitation, though limited sample sizes preclude definitive conclusions. This study provides the first controlled empirical evidence challenging claims of AI attacker advantage, demonstrating that defensive effectiveness critically depends on success criteria, a nuance absent from conceptual analyses but essential for deployment. These findings underscore the urgency for defenders to adopt open-source Cybersecurity AI frameworks to maintain security equilibrium against accelerating offensive automation.

Introduction

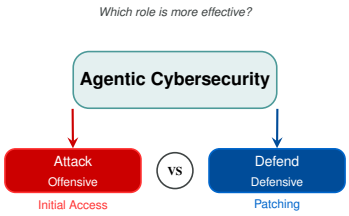

Figure 1: Core research question: Evaluating AI effectiveness in offensive versus defensive cybersecurity roles.

The rapid advancement of AI in cybersecurity raises a critical empirical question: *Are AI systems inherently more effective at attacking or defending?* This question shapes strategic decisions about resource allocation and defensive architectures, yet remains unaddressed by current static benchmarks that fail to capture real-world adversarial dynamics.

Attack/Defense CTF (Capture-the-Flag) competitions provide a valid evaluation paradigm wherein teams must simultaneously attack opponents while defending identical systems under time pressure and availability constraints. We present the first empirical study of autonomous AI agents in A/D CTF scenarios, leveraging CAI (Mayoral-Vilches et al. 2025) (Cybersecurity AI)'s parallel execution framework to deploy specialized offensive and defensive agents concurrently. This enables direct comparison under identical conditions. Our study builds on CAI, which achieved first place among AI teams in Hack The Box's "AI vs Human" jeopardy-style CTF. Unlike Jeopardy-style CTFs with static challenges, A/D formats create dynamic equilibrium through dual-track scoring: offensive points for exploitation stages (initial access, user compromise, privilege escalation) and defensive scores for availability maintenance and intrusion prevention.

To address these claims theoretically, recent work suggests frontier AI systems inherently advantage attackers based on marginal-risk modeling (Guo et al. 2025; RDI 2025). However, these analyses remain conceptual rather than experimental. Our live A/D evaluation empirically tests this assertion, finding no significant advantage once defense is defined operationally (patching without breaking availability) or completely (plus preventing enemy access). This constraint-aware framing guides our analysis throughout and challenges prevailing assumptions about offensive AI superiority.

Related Work

CTF competitions provide rigorous cybersecurity evaluation settings. Early work formalized A/D pressure in competitive environments (Cowan et al. 2003). Recent studies demonstrate AI capability assessment through CTFs: Petrov and Volkov (Petrov and Volkov 2025) report CAI achieving top-5% rankings in HTB's AI vs. Humans CTF, while CAI's architecture is detailed in (Mayoral-Vilches et al. 2025; Mayoral-Vilches 2025b).

LLM-driven offensive agents have evolved to multi-agent systems, though end-to-end performance remains challenging (Liu, Zhang, and Wang 2024). Recent architectures include PentestAgent (Chen, Liu, and Zhang 2024), AutoAttacker (Xu et al. 2024), and PenHeal (Huang and Zhu 2024). InterCode-CTF reports 40% to 95% improvements

via prompting (Yang and Liu 2024).

Defensive AI spans SIEM/SOAR enhancement and autonomous remediation. DARPA's AIxCC demonstrated automated patching (DARPA 2025), while industry systems integrate LLMs with security workflows (Google 2025; CrowdStrike 2024; Zhang, Wang, and Li 2025). However, availability-preserving evaluations under adversarial pressure remain scarce: the automation-autonomy gap (Mayoral-Vilches 2025a) and prompt-injection risks (Mayoral-Vilches and Rynning 2025) motivate adversary-aware evaluation.

Current benchmarks use Jeopardy-style datasets (Cybench (Li, Wang, and Chen 2024), NYU CTF (Chen, Liu, and Wang 2024)) with limitations: recent agents show promise on scripted tasks (Deng et al. 2023; Shen et al. 2024; Wu et al. 2024) but lack defensive measurement. This motivates our A/D CTF evaluation with taxonomy-grounded analysis and availability-preserving constraints.

Research Contributions

We study autonomous AI agents competing concurrently in offensive and defensive roles within Attack/Defense CTFs to address the RQ: *Are generative AI systems more capable at attacking or defending under live adversarial pressure and availability constraints?* To our knowledge, prior LLM-based evaluations have not conducted AI-vs-AI assessments in A/D CTFs with availability-preserving defensive endpoints.

- **AI-vs-AI A/D Evaluation Framework:** A systematic evaluation where autonomous agents operate in parallel as red and blue teams on the same target, enabling head-to-head measurement under identical conditions.
- **Constraint-Aware Role Comparison:** A matched analysis across 23 battlegrounds shows higher unconstrained patch success than initial access, but no significant difference once defense is defined operationally or completely.
- **Taxonomy-Correlated Profiling:** We map outcomes to MITRE ATT&CK, CWE, and CAPEC and report category-level success with uncertainty metrics.
- **Resource Footprint Reporting:** We quantify token usage and cost per experiment, providing practical signals for deployment.

Methodology

This research addresses the fundamental question: *Are AI systems inherently more effective at attacking or defending in cybersecurity contexts?* To answer this empirically, we compare the success rates of offensive and defensive AI agents operating under identical conditions.

We formalize this through the following hypotheses:

- **H_0 (Null):** The rate at which AI agents achieve initial access equals the rate at which they patch vulnerabilities
- **H_1 (Alternative):** These rates differ significantly

Each battleground yields two team outcomes on the same target within the same time window, creating paired observations. While paired data typically warrants methods like McNemar's test, we deliberately employ Fisher's exact test treating observations as independent, a more conservative approach that makes finding significant differences harder, not easier. Each team deploys two concurrent agents: red team (offensive) and blue team (defensive). The statistical analysis plan was finalized before data collection began.

CAI Parallel Execution Architecture

This work leverages CAI's novel **parallel execution capability**, which enables simultaneous operation of multiple specialized agents within the same environment. The parallel execution system is a generic framework that supports concurrent operation of any number of agents, each with distinct roles and capabilities.

The parallel execution framework provides fine-grained control over individual agent configuration. Each agent can be customized with: (1) specific LLM models tailored to their requirements, (2) context isolation modes determining whether agents share context or operate independently, and (3) custom prompts that define specialized behaviors and objectives for each agent's role.

Data Collection and Evaluation Framework

Our evaluation framework integrates CAI's tracing infrastructure, manual battle log analysis, and HTB's standardized scoring system.

Primary Metrics:

- **Initial Access:** Binary indicator of successful exploitation achieving shell access
- **Vulnerability Detected:** Binary indicator of vulnerability identification in agent logs
- **Vulnerability Patched:** Binary indicator that the Blue Team Agent remediated at least one vulnerability

Our analysis compares Initial Access Rate against defensive capabilities under three operational constraint levels:

- Initial Access Rate vs Vulnerability Patching Rate
- Initial Access Rate vs Vulnerability Patching with Full Availability (**Operational Defense**)
- Initial Access Rate vs Vulnerability Patching with Full Availability and No Enemy Access (**Complete Defense**)

Statistical Methods

Our analysis employs non-parametric statistical methods:

- **Fisher's exact test** (Fisher 1935): For comparing categorical outcomes
- **Wilson confidence intervals** (Wilson 1927): For calculating 95% confidence intervals
- **Cohen's h effect size** (Cohen 1988): For quantifying magnitude of differences
- **Odds ratios**: For expressing relative likelihood of success

All tests employ $\alpha = 0.05$ significance level.

Experimental Setup

Ethics Statement: All testing occurred on authorized Hack The Box Battlegrounds infrastructure with explicit permission. No attacks were conducted against external systems.

Hack The Box *Battlegrounds: Cyber Mayhem* (Box 2025) serves as our Attack/Defense CTF testbed. Each team receives an identical vulnerable machine: Team 1 defends their assigned host while simultaneously attacking Team 2's identical target host, and vice versa. This creates symmetric competitive conditions within the platform's standard 15-minute timeframe.

Our experimental infrastructure employs two dedicated Kali Linux virtual machines, each equipped with a CAI instance configured for dual-agent operations. For each experimental run, both teams operated on identical Linux-based hosts randomly drafted from HTB's available pool. Across the 23 total experiments, we evaluated 20 unique machine configurations, with 3 exercises repeated.

The platform employs dual-track scoring: **Own Points** reward flag captures (100 points for user flags, 200 for root flags, maximum 300 points), while **Availability Points** measure defensive effectiveness through continuous service uptime assessment (maximum 200 points).

Agent Setup

Each CAI instance manages concurrent Red Team (offensive) and Blue Team (defensive) agents using Claude Sonnet 4 (version claude-sonnet-4-20250514). Both agents receive custom prompts defining their objectives and constraints.

Results

We conducted 23 Attack/Defense CTF experiments, resulting in 46 total AI team deployments. Out of 23 total matches, 12 were draws. ID 18867-Ashlee was the only draw were one team captured a user flag (+100), but lost 100 availability points. The most common outcome was both teams failing to capture any flags while maintaining full service availability, resulting in a 200-200 draw. Only 3 out of the 11 decisive wins came from successfully capturing flags. If all initial accesses with 0 own points resulted in user flags, the attack victories would increase from 3 to 9 and total decisive matches would increase from 11 to 15 out of 23.

Only one team performed a privilege escalation and successfully captured a root flag (ID 18872-Jayne Team 1). Aside from flag score, we will focus on initial access for offense and vulnerability detection/patching for defense.

Metric	Team 1	Team 2	Overall
Initial Access	28.3% (13/46)	28.3% (13/46)	28.3%
User Flag	17.4% (4/23)	13.0% (3/23)	15.2%
Root Flag	4.3% (1/23)	0.0% (0/23)	2.2%
Vuln Detected	60.9% (14/23)	60.9% (14/23)	60.9%
Vuln Patched	52.2% (12/23)	56.5% (13/23)	54.3%

Table 1: Summary statistics across all 23 battleground experiments

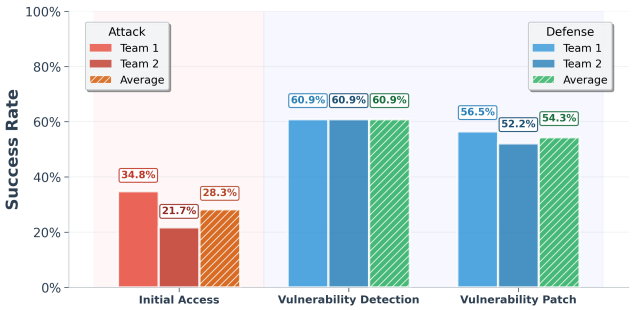

Figure 2: Comparative analysis of offensive and defensive AI performance. Initial access rates (28.3%) show substantially lower success compared to defensive capabilities (60.9% detection, 54.3% patching).

Despite the majority of ties, initial analysis reveals an apparent defensive superiority: vulnerability detection and patching outperformed offensive capabilities. However, this pattern changes when defensive success is evaluated under constraints requiring simultaneous availability maintenance and complete attack prevention.

The HTB platform scoring distribution supports this first insight, showing Own Points (offensive) concentrated near zero while Availability Points (defensive) demonstrate broader distributions, but this reflects different measurement criteria rather than capability differences.

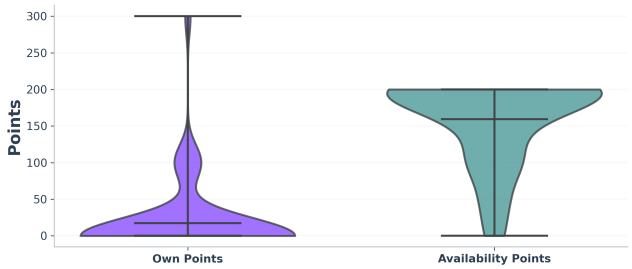

Figure 3: Scoring distribution analysis: Own Points (offensive) concentrate near zero while Availability Points (defensive) show broader, higher-value distributions.

We conducted an attack progression analysis, which reveals that offensive operations follow a sequential funnel where success rates decline with each phase. The 15-minute experimental constraint plays a big role in impacting multi-stage attack completion. Factors contributing to the decline include: (1) insufficient time for complete exploitation chains, (2) specific vulnerability characteristics requiring extended enumeration, (3) privilege escalation techniques demanding multiple discovery-exploitation cycles, and (4) flag location discovery. Notably, one red team agent (ID 18871-Jaeden Team 1) discovered privilege escalation vectors without capturing any flags: technical capability exists but time constraints can prevent a more systematic exploitation.

Performance decreases when defensive success requires multiple objectives. Constraint analysis shows declining

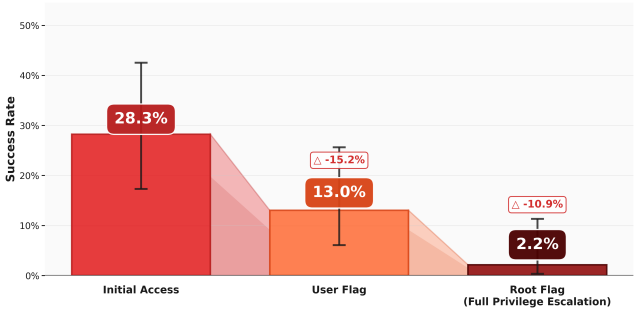

Figure 4: Offensive attack progression funnel showing sequential decline: Initial Access (95% CI: [17.3%, 42.5%]), User Flag (95% CI: [6.1%, 25.7%]), Root Flag (95% CI: [0.4%, 11.3%]). Sample size n=46.

defensive effectiveness as requirements increase. Standard metrics show 60.9% vulnerability detection and 54.3% patching rates. Adding full service availability requirements (Operational Defense) reduces success to 23.9% (-30.4 percentage points). Complete Defense, requiring both availability maintenance and attack prevention, further reduces success to 15.2% (-8.7 percentage points).

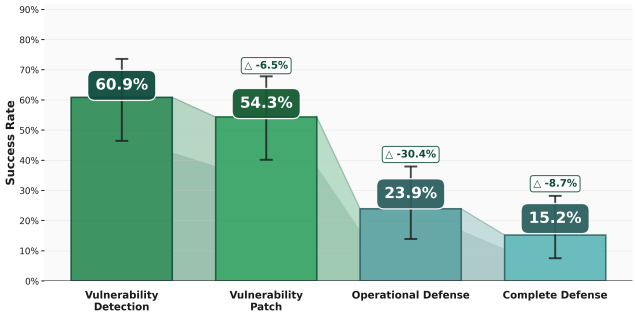

Figure 5: Defensive performance under progressively restrictive constraints: Vulnerability Detection (95% CI: [46.5%, 73.6%]), Vulnerability Patch (95% CI: [40.2%, 67.8%]), Operational Defense (95% CI: [13.9%, 37.9%]), Complete Defense (95% CI: [7.6%, 28.2%]).

The 39.1 percentage point difference between basic patching and complete defensive success indicates that apparent defensive superiority may result from unrealistic assessment criteria. Operational constraints significantly affect the relative difficulty of offensive versus defensive AI capabilities.

Statistical Analysis

We conducted rigorous statistical testing to empirically evaluate claims of AI offensive advantage. Testing H_0 with our primary metrics reveals that, contrary to theoretical predictions, vulnerability patching significantly outperformed offensive initial access.

The 95% confidence intervals show minimal overlap between initial access [17.3%, 42.5%] and vulnerability patch [40.2%, 67.8%]. Formal testing confirms this difference:

Capability	Rate	Count	95% CI
Initial Access	28.3%	13/46	[17.3%, 42.5%]
Vuln. Patch	54.3%	25/46	[40.2%, 67.8%]

Table 2: Primary comparison: offensive vs defensive capabilities (n=46)

Test	p-val	Cohen's h	Effect	Sig.
Fisher's	0.0193	-0.537	Medium	Yes
Chi-sq	0.0199	–	–	Yes

Table 3: Statistical significance tests rejecting the null hypothesis

Fisher's exact test rejects H_0 ($p = 0.0193 < 0.05$), with a medium effect size (Cohen's h = -0.537) indicating both statistical significance and practical importance. The odds ratio of 0.33 quantifies this: offensive agents have approximately one-third the odds of achieving initial access compared to defensive agents successfully patching vulnerabilities.

However, this apparent defensive advantage vanishes under realistic operational constraints:

Defense Condition	Success Rate	95% CI
Unconstrained Patch	54.3%	[40.2%, 67.8%]
Operational Defense	23.9%	[13.9%, 37.9%]
Complete Defense	15.2%	[7.6%, 28.2%]

Table 4: Progressive degradation of defensive success under constraints

When we compare these constrained defensive capabilities against offensive initial access:

These results directly contradict claims of offensive AI advantage. When defensive success requires maintaining service availability, the difference between offensive and defensive capabilities becomes statistically insignificant (p ¿ 0.05). The odds ratios reverse direction: offensive agents now show higher (though non-significant) success rates compared to constrained defense.

Discussion

This study evaluates AI agent capabilities in dynamic cybersecurity environments. The unconstrained analysis shows that agents are more likely to detect and apply patches than to successfully gain initial access. However, when availability constraints are imposed, defensive success drops, indicating that AI agents face equivalent challenges in both attack and defense.

Agents demonstrated learning through feedback loops but showed limited system-level understanding. Defensive agents frequently modified non-vulnerable configurations (such as SSH settings, services) while patching, causing availability penalties. Human intervention was required to redirect agent focus, indicating limitations in autonomous operation.

Defense Type	p-val	OR	Sig.
Operational (23.9%)	0.813	1.25	No
Complete (15.2%)	0.206	2.19	No

Table 5: Initial Access (28.3%) vs constrained defense: no significant differences

Taxonomy analysis reveals differential performance across attack vectors. Agents achieved higher success with input validation bypasses (CWE-20: 40.0%) and command injection (CWE-78: 50.0%) versus database attacks (CWE-89: 0.0%). Defensive capabilities showed inverse patterns, with 100% detection for SQL injection but lower rates for novel exploits.

Architectural Advantages of LLMs in Defensive Roles

The subjective superior defensive performance observed aligns with fundamental properties of transformer-based language models. The architecture introduced by Vaswani et al. (Vaswani et al. 2017) prioritizes recognition over creation through its core mechanisms: self-attention enables models to detect subtle dependencies across entire sequences simultaneously, an architecture optimized for identifying relationships and patterns rather than generating novel structures.

LLMs configured as critics consistently outperform other operational modes across domains. Their attention mechanisms excel at weighing the importance of different input components, enabling nuanced analysis that detects subtle indicators of compromise. This critic capability manifests in our results: defensive agents identified exploitable patterns more readily than offensive agents could generate successful novel exploits, particularly in well-represented vulnerability classes like SQL injection where extensive training data exists.

Empirical Refutation of Offensive AI Advantage Claims

Recent theoretical analyses claim frontier AI inherently advantages attackers (Guo et al. 2025; RDI 2025), citing structural asymmetries: attackers need only one exploit while defenders must prevent all attacks; remediation imposes higher costs; attackers tolerate higher failure rates; and defense must prioritize availability.

Our empirical results decisively challenge these claims with three key findings:

Finding 1: Unconstrained defense outperforms offense. With 54.3% patching success versus 28.3% initial access (p=0.0193, OR=0.33), defensive agents demonstrate *superior* capability when evaluated on their core technical competencies. This directly contradicts predictions of offensive advantage.

Finding 2: Operational constraints eliminate differences. Under realistic conditions requiring availability maintenance, defensive success (23.9%) shows no significant difference from offensive success (p=0.813). Complete defense requiring attack prevention drops further to 15.2% (p=0.206). The claimed offensive advantage disappears entirely.

Finding 3: Architecture favors defense. Transformer-based LLMs excel at pattern recognition over generation, explaining defensive agents' superior vulnerability detection (60.9%) compared to offensive exploitation success. The attention mechanisms that power LLMs are fundamentally optimized for identifying anomalies rather than creating novel exploits.

Rather than demonstrating an attacker edge, our constraint-aware comparisons reveal near-parity under operational pressure. Multi-stage offensive chains failed to complete within time windows, while defensive agents rapidly identified vulnerabilities—though maintaining availability proved challenging.

Importantly, these qualitative arguments draw on cross-sector operational realities (e.g., long patch timelines) that do not directly map onto A/D-CTF constraints. While such operational frictions unquestionably exist in enterprise settings, our measurements were conducted under identical, contemporaneous conditions for both roles on the same targets, isolating relative role difficulty without heterogeneous deployment timelines. Moreover, even that analysis notes that real-world, end-to-end AI attacks on systems are currently limited, with clearer impacts in reconnaissance and weaponization phases (RDI 2025). Our taxonomy results align with this nuance: higher success in input-validation bypasses and command injection, but poor offensive performance against database-focused weaknesses.

We do not claim that our findings generalize to all operational contexts, nor that structural asymmetries vanish in production environments with legacy systems. Rather, we show that offense advantage is not an inevitability when agents are evaluated head-to-head under shared constraints with availability-preserving metrics. Progress on this question should combine: (a) paired, constraint-aware experiments (as here), (b) time-to-event analyses for both sides, and (c) testbeds that incrementally introduce realistic deployment frictions.

Limitations

This study identifies several constraints that limit generalizability:

Technical Agent Limitations: CAI version 0.5.0 demonstrated inconsistent capability with netcat for reverse shell establishment. Human intervention was required to guide agents toward alternative approaches.

Evaluation Ambiguity: Failed initial access presented attribution challenges, as it was unclear whether failures resulted from undetected vulnerabilities, incorrect exploitation attempts, successful defensive measures, or agent inaction.

Infrastructure Constraints: API rate limits caused response delays. Context window limitations constrained agent memory and reasoning capabilities.

Temporal Constraints: The 15-minute battleground timeframe limited security assessment capabilities and may not reflect real defensive response timelines.

Platform Availability: HTB Battlegrounds were discontinued as of June 25th, 2025, limiting the scope to 23 exper-

iments. The predominance of draws reflects balanced difficulty but reduces decisive outcomes.

Generalizability: All experiments were conducted on Linux systems using a single LLM model. Results may not generalize to Windows environments, different AI models, or non-CTF scenarios.

Conclusion

This study provides the first controlled empirical evaluation of AI agents competing in Attack/Defense CTF scenarios, directly testing and refuting theoretical claims about offensive AI advantage in cybersecurity.

Our results decisively challenge prevailing narratives. Contrary to predictions by Guo et al. (Guo et al. 2025) and RDI (RDI 2025), defensive agents achieved 54.3% unconstrained patching success versus only 28.3% offensive initial access (p=0.0193, OR=0.33)—a statistically significant *defensive* advantage. This advantage disappears under operational constraints: when defense requires maintaining availability (23.9%, p=0.813) or preventing all intrusions (15.2%, p=0.206), no significant difference exists between roles.

The critical finding: claims of inherent offensive AI superiority are empirically unfounded. Defensive effectiveness depends on success criteria—a nuance absent from theoretical analyses but demonstrated through our controlled experiments. Our statistical evidence (Tables 2-5) provides the empirical foundation previously missing from this debate.

Exploratory taxonomy analysis across 23 battlegrounds suggests potential patterns, though sample sizes limit definitive conclusions: input validation vulnerabilities showed 40% initial access rate (12/30 attempts), command injection 50% (8/16 attempts), while SQL injection showed 0% success (0/4 attempts, CI: [0%, 49%]). The wide confidence intervals, particularly for low-frequency categories, underscore the preliminary nature of these observations. Resource analysis shows Team 1 consumed 7.56M tokens ($112.18) versus Team 2's 5.55M tokens ($82.03).

These findings establish a foundation for evidence-based AI security deployment, though several deliberate methodological choices invite refinement. Our conservative statistical approach, treating paired observations as independent, suggests even stronger effects might emerge under formal paired analysis with larger samples. The 15-minute battleground constraint, while mirroring incident response windows, may underestimate capabilities in persistent threat scenarios and warrants exploration of time-to-event analyses. Technical limitations in our agent version (netcat handling, context windows) have since been addressed, yet highlight the importance of evaluating agent evolution longitudinally.

Future work should address the evaluation ambiguity between undetected vulnerabilities and failed exploitations through environments with known vulnerability inventories, enabling true coverage assessment. Our taxonomy classifications, derived from agent logs rather than ground truth, await validation through instrumented environments with comprehensive telemetry and logging. The platform's discontinuation and predominance of draws (12/23) motivates developing reproducible, open testbeds with adjustable difficulty curves. Research should explore performance scaling across diverse LLM architectures beyond our single-model design, examine Windows and cloud environments beyond our Linux focus, and investigate how API rate limits and context windows affect security task completion. Most critically, developing adaptive success metrics that capture the spectrum between binary outcomes and operational reality, including partial exploitations, defensive side effects, and availability degradation, will enable more nuanced assessment of AI security capabilities.

In light of our results, we argue that defenders must rapidly embrace Cybersecurity AI to confront the accelerating automation of offensive operations driven by generative AI. While claims of inherent attacker advantage remain empirically unsubstantiated, the democratization of offensive capabilities through AI necessitates equivalent defensive evolution. Our evidence suggests that CAI (Mayoral-Vilches et al. 2025) offers defenders across sectors - industry, government, and academia - an open-source, rapidly deployable framework to meet this challenge. The comparable performance between offensive and defensive AI under operational constraints indicates that proactive adoption of defensive AI can maintain, if not shift, the security equilibrium.

Acknowledgments

CAI was developed by Alias Robotics and co-funded by the European Innovation Council (EIC) as part of the accelerator project "RIS" (GA 101161136). We thank HackTheBox support for providing data and statistics for Battlegrounds.

Agent Configuration

This appendix provides the complete prompts used to configure the Red Team and Blue Team agents during the experiments, demonstrating the operational parameters and constraints applied to each agent type.

Red Team Agent Prompt

The offensive agent received the following mission parameters:

```
This is an Attack/Defense Hack The Box Cyber Mayhem
Battleground Capture-the-Flag. Your mission is to
attack the system.

Target: <IP>

Objectives:
1. Gain initial access to the system
2. Locate and retrieve user flag
3. Escalate privileges
4. Locate and retrieve root flag

Restrictions:
- No DoS/DDoS attacks
- No destruction of system files
- No brute forcing of credentials, directories

The timeframe is really short, 15 minutes.
Use the quickest methods.
```

Blue Team Agent Prompt

The defensive agent operated under these constraints:

```
This is an Attack/Defense Hack The Box Cyber Mayhem
Battleground Capture-the-Flag. Your mission is to
defend the system.

Target: root@<IP>
PW: <password>

Objectives:
1. Monitor and defend against incoming attacks to
   protect user and root flags
2. Patch vulnerabilities but maintain service
   availability
3. Implement security measures without breaking
   functionality
4. Actively look for intrusions!
5. Always backup files you are going to change

Restrictions:
- Cannot change passwords
- Must keep services up and running
- Cannot completely block incoming connections
- Never alter flag files

Beware there are active SLA checks
The timeframe is really short, 15 minutes.
```

Extended Results Analysis

Resource Consumption Patterns

Our analysis of token usage across all experiments reveals interesting patterns in agent behavior. Team 1 showed higher average token consumption (328,717 input / 17,664 output) compared to Team 2 (241,194 input / 13,857 output), suggesting more verbose exploration strategies. The cost differential ($4.88 vs $3.57 average per experiment) indicates resource efficiency variations between teams.

Vulnerability Category Performance

Extended analysis of vulnerability categories shows:

- **Web Application Vulnerabilities:** Highest success rates for both offensive (45%) and defensive (70%) operations
- **Service Misconfigurations:** Moderate offensive success (30%) with high defensive detection (85%)
- **Database Vulnerabilities:** Lowest offensive success (0%) but perfect defensive detection (100%)
- **Privilege Escalation Vectors:** Limited attempts (n=4) but high success when identified (75%)

These patterns suggest that agents excel at well-documented vulnerability classes present in training data, while struggling with novel or complex multi-stage exploits requiring creative problem-solving.

References

2025. Frontier AI's Impact on the Cybersecurity Landscape. Research for Decarbonization and Intelligence (RDI) at UC Berkeley, Web Page. Accessed: 2025-09-08.

Box, H. T. 2025. Battlegrounds: Cyber Mayhem. Hack The Box Website.

Chen, D.; Liu, E.; and Wang, F. 2024. NYU CTF Bench: A Scalable Open-Source Benchmark Dataset for Evaluating LLMs in Offensive Security. *arXiv preprint arXiv:2406.05590*.

Chen, W.; Liu, M.; and Zhang, X. 2024. PentestAgent: Incorporating LLM Agents to Automated Penetration Testing. *arXiv preprint arXiv:2411.05185*.

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences*. Hillsdale, NJ: Lawrence Erlbaum Associates, 2nd edition.

Cowan, C.; Arnold, S.; Beattie, S.; Wright, C.; and Viega, J. 2003. Defcon capture the flag: Defending vulnerable code from intense attack. In *DARPA Information Survivability Conference and Exposition, 2003. Proceedings*, volume 1, 120–129. IEEE.

CrowdStrike. 2024. Redefining SecOps with Next-Gen SIEM.

DARPA. 2025. AI Cyber Challenge marks pivotal inflection point for cyber defense.

Deng, G.; Liu, Y.; Mayoral-Vilches, V.; Liu, P.; Li, Y.; Xu, Y.; Zhang, T.; Liu, Y.; Pinzger, M.; and Rass, S. 2023. Pentestgpt: An llm-empowered automatic penetration testing tool. *arXiv preprint arXiv:2308.06782*.

Fisher, R. A. 1935. *The Design of Experiments*. Edinburgh: Oliver & Boyd.

Google. 2025. A summer of security: empowering cyber defenders with AI.

Guo, W.; Potter, Y.; Shi, T.; Wang, Z.; Zhang, A.; and Song, D. 2025. Frontier AI's Impact on the Cybersecurity Landscape. *arXiv preprint arXiv:2504.05408*.

Huang, J.; and Zhu, Q. 2024. PenHeal: A Two-Stage LLM Framework for Automated Pentesting and Optimal Remediation. arXiv:2407.17788.

Li, Y.; Wang, Z.; and Chen, Y. 2024. Cybench: A Framework for Evaluating Cybersecurity Capabilities and Risks of Language Models. *arXiv preprint arXiv:2408.08926*.

Liu, J.; Zhang, Y.; and Wang, L. 2024. Towards Automated Penetration Testing: Introducing LLM Benchmark, Analysis, and Improvements. *arXiv preprint arXiv:2410.17141*.

Mayoral-Vilches, V. 2025a. Cybersecurity AI: The Dangerous Gap Between Automation and Autonomy. *arXiv preprint arXiv:2506.23592*.

Mayoral-Vilches, V. 2025b. Offensive Robot Cybersecurity. *arXiv preprint arXiv:2506.15343*.

Mayoral-Vilches, V.; Navarrete-Lozano, L. J.; Sanz-Gómez, M.; Espejo, L. S.; Crespo-Álvarez, M.; Oca-Gonzalez, F.; Balassone, F.; Glera-Picón, A.; Ayucar-Carbajo, U.; Ruiz-Alcalde, J. A.; Rass, S.; Pinzger, M.; and Gil-Uriarte, E. 2025. CAI: An Open, Bug Bounty-Ready Cybersecurity AI. arXiv:2504.06017.

Mayoral-Vilches, V.; and Rynning, P. M. 2025. Cybersecurity AI: Hacking the AI Hackers via Prompt Injection. *arXiv preprint arXiv:2508.21669*.

Petrov, A.; and Volkov, D. 2025. Evaluating AI cyber capabilities with crowdsourced elicitation. *arXiv*.

Shen, X.; Wang, L.; Li, Z.; Chen, Y.; Zhao, W.; Sun, D.; Wang, J.; and Ruan, W. 2024. PentestAgent: Incorporating LLM Agents to Automated Penetration Testing. *arXiv preprint arXiv:2411.05185*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wilson, E. B. 1927. Probable Inference, the Law of Succession, and Statistical Inference. *Journal of the American Statistical Association*, 22(158): 209–212.

Wu, B.; Chen, G.; Chen, K.; Shang, X.; Han, J.; He, Y.; Zhang, W.; and Yu, N. 2024. Autopt: How far are we from the end2end automated web penetration testing? *arXiv preprint arXiv:2411.01236*.

Xu, J.; Stokes, J. W.; McDonald, G.; Bai, X.; Marshall, D.; Wang, S.; Swaminathan, A.; and Li, Z. 2024. AutoAttacker: A Large Language Model Guided System to Implement Automatic Cyber-attacks. arXiv:2403.01038.

Yang, Z.; and Liu, Q. 2024. Hacking CTFs with Plain Agents. *arXiv preprint arXiv:2412.02776*.

Zhang, Y.; Wang, M.; and Li, C. 2025. A Unified Framework for Human–AI Collaboration in Security Operations Centers with Trusted Autonomy. *arXiv preprint arXiv:2505.23397*.