CTFers: LLMs that Learn to Hack and Defend

Song Xixuan*
Department of Computer Science
Tsinghua University
songxx21@mails.tsinghua.edu.cn

Abstract

Capture-the-Flag (CTF) competitions offer a challenging benchmark for testing AI reasoning in cybersecurity. However, large language models (LLMs) still perform far below expert level. This project proposes CTFers, a reinforcement learning framework that trains LLMs to autonomously solve CTF challenges and compete against other LLMs in adversarial settings. Using structured feedback, sandboxed environments, and Generalized Rehearsal Policy Optimization (GRPO), CTFers enables models to iteratively refine exploitation and defense strategies. Our goal is to bridge the gap between static prompt-based reasoning and adaptive cyber reasoning, advancing toward self-improving AI agents capable of mastering both offensive and defensive security tasks.

1 Introduction

Capture-the-Flag (CTF) competitions have long been recognized as a gold standard for assessing problem-solving and reasoning skills in cybersecurity [10, 18]. Each CTF challenge requires a solver to understand complex system behavior, analyze vulnerabilities, craft exploits, or design countermeasures. These skills require the use of analytical reasoning and practical tools. Recent studies such as NYU CTF Bench [10] and CyBench [16] have begun to quantify how well large language models (LLMs) can perform in this domain. Despite impressive progress in reasoning and coding, even state-of-the-art models like Claude 4.5 [2] achieve only around 55% accuracy, far below human experts [16].

Most prior work evaluates LLMs through static prompting or chain-of-thought reasoning [15, 8], where models passively generate answers to pre-defined problems. However, cybersecurity reasoning is inherently interactive and adversarial: agents must iteratively test hypotheses, execute tools, and adapt to dynamic feedback. This gap calls for a new paradigm that allows LLMs not only to solve CTF challenges but also to learn from their interactions with the environment and even compete against other agents.

To address this, we propose **CTFers**, a reinforcement-learning-based framework that trains LLMs to autonomously hack and defend in realistic CTF environments. CTFers integrates Generalized Rehearsal Policy Optimization (GRPO) [12] to enable stable on-policy learning from structured feedback signals such as exploit success or defense robustness. The system supports both single-agent learning—where an LLM iteratively improves by interacting with sandboxed challenges—and multi-agent adversarial play, where two LLMs compete in attack-versus-defense scenarios based on environments such as CAGE [3, 4] or CybORG [3]. By combining imitation learning from CTF traces [18] and RL-based fine-tuning, CTFers aims to bridge the gap between static reasoning and adaptive cyber intelligence.

Through experiments on CTF-Dojo [18], NYU CTF Bench [10], and CyBench [16], this work seeks to demonstrate the emergence of self-improving cyber reasoning behaviors in LLMs. Ultimately, we

*https://github.com/songxxzp

Preprint.

envision CTFers as a step toward autonomous AI security agents that can continuously learn to hack, defend, and evolve in the face of novel digital threats.

2 Problem Definition

We formulate CTF solving as a sequential decision-making problem in a reinforcement learning (RL) framework. At each timestep t, the agent interacts with an environment representing a cybersecurity challenge.

2.1 Single-Agent CTF Solving

Let $\mathcal{E}$ denote a CTF environment, defined by a tuple

$$\mathcal{E} = (\mathcal{S}, \mathcal{A}, P, R, \gamma),$$

where $\mathcal{S}$ is the set of states (e.g., challenge descriptions, command-line outputs, or execution traces), $\mathcal{A}$ is the action space (e.g., textual commands, code snippets, or reasoning steps), $P(s_{t+1}|s_t, a_t)$ is the environment transition function, $R(s_t, a_t)$ is the reward function, and γ is the discount factor.

At each step, the LLM policy $\pi_\theta(a_t|s_t)$ generates an action conditioned on the observed state. The objective of the agent is to maximize the expected cumulative reward:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right],$$

where the reward R is defined according to the task progress. Here is an example:

$$R(s_t, a_t) = \begin{cases} +1, & \text{if the flag is successfully captured;} \\ +0.1, & \text{if partial progress is made (e.g., vulnerability identified);} \\ -0.01, & \text{for invalid actions or crashes.} \end{cases}$$

This formulation allows the LLM to iteratively refine its reasoning and exploitation strategy through structured feedback.

2.2 Multi-Agent Adversarial Setting

Beyond single-agent solving, CTFers extends to a two-agent adversarial environment

$$\mathcal{E}^{(2)} = (\mathcal{S}, \mathcal{A}_A, \mathcal{A}_D, R_A, R_D),$$

where an *attacker* LLM (π_{θ_A}) attempts to exploit vulnerabilities, and a *defender* LLM (π_{θ_D}) seeks to patch or mitigate them. The game is competitive, with opposite reward functions:

$$R_A = -R_D, \quad R_A(s_t, a_t) = \begin{cases} +1, & \text{if exploit succeeds;} \\ 0, & \text{if attack is blocked.} \end{cases}$$

Training proceeds alternately, allowing both agents to improve via reinforcement learning using the same GRPO objective:

$$\theta \leftarrow \theta + \eta \nabla_\theta J(\theta),$$

where η is the learning rate.

2.3 Training Objectives

Formally, the goal of **CTFers** is to learn policies

$$\pi^*_{\theta_A}, \pi^*_{\theta_D} = \arg\max_{\pi_{\theta_A}} \mathbb{E}[R_A], \quad \arg\max_{\pi_{\theta_D}} \mathbb{E}[R_D],$$

such that both attacker and defender agents evolve toward more sophisticated, self-improving cybersecurity reasoning behaviors across diverse CTF tasks.

3 Related Work

3.1 LLMs for Cybersecurity and CTF Challenges

The use of large language models (LLMs) in cybersecurity, particularly for Capture-the-Flag (CTF) competitions, has attracted increasing attention. Early works evaluate the ability of LLMs to tackle security puzzles via prompting. For instance, Shao et al. [9] present an empirical evaluation of multiple LLMs on CTF tasks, showing that while LLMs surpass average human participants under certain settings, significant gaps remain in autonomous challenge solving. Tann et al. [13] investigate LLM performance on CTF challenges and certification-questions, revealing limitations in complex vulnerability exploitation and multi-step reasoning. More recently, frameworks such as CRAKEN [11] propose knowledge-based LLM agents for cybersecurity tasks, and Cyber-Zero [19] synthesizes agent trajectories for training cybersecurity agents without runtime instrumentation. On the benchmark side, the creation of specialized CTF / crypto datasets such as AICrypto [14] further enables rigorous assessments of LLMs' security-reasoning capabilities. These works demonstrate the promise of LLMs in security domains, but largely rely on static evaluation or prompting; they rarely incorporate iterative environment interaction, feedback loops or adversarial agent setups.

3.2 Reinforcement Learning for LLM Reasoning and Agentic Behavior

Parallel to the domain-specific work on cybersecurity, there is a rich line of research applying reinforcement learning (RL) and agentic methods to LLMs in broader reasoning, planning and tool-use contexts. For example, Hao et al. [5] introduce *RL of Thoughts* (RLoT), where a lightweight RL navigator selects logic blocks at inference time to enhance LLM reasoning on benchmarks such as AIME or MATH. Surveys such as Zhang et al. [17] and Liu et al. [7] comprehensively review RL methods for large reasoning models and LLMs' lifecycle respectively. Moreover, empirical studies like Jin et al. [6] examine RL for reasoning-search interleaved agents, highlighting key design choices around reward shaping, action design, and tool invocation. These works show that RL methods are increasingly used to refine LLM behavior beyond supervised/few-shot prompting; however, applications in adversarial interactive domains (such as attacker-defender CTFs) remain under-explored.

4 Proposed Method

4.1 Motivation

While recent advances[16, 10, 19] have established both the performance of LLMs on static CTF benchmarks and the benefits of reinforcement learning (RL) for reasoning, yet there remains a clear gap at their intersection: little work explores multi-agent attacker–defender dynamics, or the integration of sandboxed environment interaction with specialized policy optimization.

To bridge this gap, we propose **CTFers**, a reinforcement-learning-based framework that integrates LLMs with interactive CTF environments, supporting both single-agent challenge solving and multi-agent adversarial play. Our goal is to move beyond static prompting toward self-improving cyber agents capable of hacking and defending adaptively in dynamic, feedback-rich settings.

4.2 Datasets and Benchmarks

We plan to train and evaluate CTFers on the following datasets and benchmark suites:

- **Training datasets:**
  - **CTF-Dojo[18]** – contains 658 CTF challenges and 486 traces, suitable for imitation pretraining and environment construction.
  - **Cyber-Zero[19]** – a large-scale dataset with tool-based interactions and multi-step challenge traces.
- **Evaluation benchmarks:**
  - **NYU CTF Bench[10]** – a public leaderboard for LLM CTF solving performance.

– **CyBench[16]** – the most recent and challenging benchmark, where current top model Claude 4.5 achieve about 55%.

For adversarial experiments, we will also use RL-based simulation environments such as the **CAGE Challenge series**[3], where attacker and defender agents compete in networked cyber arenas.

4.3 Baseline Approaches

We will compare CTFers against several strong baselines:

- State-of-the-art LLMs (GPT-5 and Claude 4.5) using prompting and reasoning.
- **EnIGMA** [1] – an agentic LLM framework combining reasoning and tool-use, shown to outperform static prompting on CTF tasks.
- Standard reinforcement-learning baselines such as PPO and GRPO variants, as used in prior cyber defense simulations like CAGE [3].

4.4 Method Implementation

CTFers is designed in two stages: a **Single-Agent Solving Phase** followed by a **Multi-Agent Adversarial Phase**. Both phases embed LLM policies within reinforcement learning loops in sandboxed CTF environments.

Single-Agent Solving Phase.

1. **Environment construction:** Each challenge is wrapped as a Gym-style interface $\mathcal{E} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$. States s_t include challenge descriptions, logs, and runtime outputs; actions a_t correspond to reasoning steps, tool calls, or code snippets.
2. **Imitation pretraining:** Using traces from CTF-Dojo to initialize the policy $\pi_\theta(a \mid s)$ via supervised learning, improving early exploration efficiency.
3. **RL fine-tuning:** We then apply Generalized Rehearsal Policy Optimization (GRPO) to maximize
$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_t \gamma^t R(s_t, a_t)\right],$$
allowing stable on-policy learning while retaining useful trajectories.
4. **Evaluation:** The resulting model is evaluated on NYU CTF Bench and CyBench for solve rate, sample efficiency, and reward convergence.

Multi-Agent Adversarial Phase.

1. **Arena setup:** We adopt two interacting agents—an attacker π_{θ_A} and defender π_{θ_D}—within CAGE/CybORG. Attacker actions include reconnaissance, exploitation, and lateral movement; defender actions include patching and isolation.
2. **Training:** We alternately train π_{θ_A} and π_{θ_D} using on-policy GRPO, periodically freezing one to encourage adversarial adaptation and co-evolution.
3. **Generalization:** Both agents are evaluated on unseen network topologies and opponent strategies to assess robustness.

References

[1] Talor Abramovich, Meet Udeshi, Minghao Shao, Kilian Lieret, Haoran Xi, Kimberly Milner, Sofija Jancheska, John Yang, Carlos E. Jimenez, Farshad Khorrami, Prashanth Krishnamurthy, Brendan Dolan-Gavitt, Muhammad Shafique, Karthik Narasimhan, Ramesh Karri, and Ofir Press. Enigma: Interactive tools substantially assist lm agents in finding security vulnerabilities, 2025. URL https://arxiv.org/abs/2409.16165.

[2] Anthropic. Claude sonnet 4.5 system card. Technical report, Anthropic, 2025. URL https://assets.anthropic.com/m/12f214efcc2f457a/original/Claude-Sonnet-4-5-System-Card.pdf. Accessed November 11, 2025.

[3] Callum Baillie, Maxwell Standen, Jonathon Schwartz, Michael Docking, David Bowman, and Junae Kim. Cyborg: An autonomous cyber operations research gym, 2020. URL https://arxiv.org/abs/2002.10667.

[4] Sebastián R. Castro, Roberto Campbell, Nancy Lau, Octavio Villalobos, Jiaqi Duan, and Alvaro A. Cardenas. Large language models are autonomous cyber defenders. In *2025 IEEE Conference on Artificial Intelligence (CAI)*, page 1125–1132. IEEE, May 2025. doi: 10.1109/cai64502.2025.00195. URL http://dx.doi.org/10.1109/CAI64502.2025.00195.

[5] Qianyue Hao, Sibo Li, Jian Yuan, and Yong Li. Rl of thoughts: Navigating llm reasoning with inference-time reinforcement learning, 2025. URL https://arxiv.org/abs/2505.14140.

[6] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

[7] Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. Reinforcement learning meets large language models: A survey of advancements and applications across the llm lifecycle, 2025. URL https://arxiv.org/abs/2509.16679.

[8] OpenAI. Openai o1: Emergent reasoning from reinforcement learning. *Technical Report*, 2024.

[9] Minghao Shao, Boyuan Chen, Sofija Jancheska, Brendan Dolan-Gavitt, Siddharth Garg, Ramesh Karri, and Muhammad Shafique. An empirical evaluation of llms for solving offensive security challenges, 2024. URL https://arxiv.org/abs/2402.11814.

[10] Minghao Shao, Sofija Jancheska, Meet Udeshi, Brendan Dolan-Gavitt, Haoran Xi, Kimberly Milner, Boyuan Chen, Max Yin, Siddharth Garg, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, and Muhammad Shafique. Nyu ctf bench: A scalable open-source benchmark dataset for evaluating llms in offensive security, 2025. URL https://arxiv.org/abs/2406.05590.

[11] Minghao Shao, Haoran Xi, Nanda Rani, Meet Udeshi, Venkata Sai Charan Putrevu, Kimberly Milner, Brendan Dolan-Gavitt, Sandeep Kumar Shukla, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri, and Muhammad Shafique. Craken: Cybersecurity llm agent with knowledge-based execution, 2025. URL https://arxiv.org/abs/2505.17107.

[12] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[13] Wesley Tann, Yuancheng Liu, Jun Heng Sim, Choon Meng Seah, and Ee-Chien Chang. Using large language models for cybersecurity capture-the-flag challenges and certification questions, 2023. URL https://arxiv.org/abs/2308.10443.

[14] Yu Wang, Yijian Liu, Liheng Ji, Han Luo, Wenjie Li, Xiaofei Zhou, Chiyun Feng, Puji Wang, Yuhan Cao, Geyuan Zhang, Xiaojian Li, Rongwu Xu, Yilei Chen, and Tianxing He. Aicrypto: A comprehensive benchmark for evaluating cryptography capabilities of large language models, 2025. URL https://arxiv.org/abs/2507.09580.

[15] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

[16] Andy K. Zhang, Neil Perry, Riya Dulepet, Joey Ji, Celeste Menders, Justin W. Lin, Eliot Jones, Gashon Hussein, Samantha Liu, Donovan Jasper, Pura Peetathawatchai, Ari Glenn, Vikram Sivashankar, Daniel Zamoshchin, Leo Glikbarg, Derek Askaryar, Mike Yang, Teddy Zhang, Rishi Alluri, Nathan Tran, Rinnara Sangpisit, Polycarpos Yiorkadjis, Kenny Osele, Gautham Raghupathi, Dan Boneh, Daniel E. Ho, and Percy Liang. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models, 2025. URL https://arxiv.org/abs/2408.08926.

[17] Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiaze Ma, Xuekai Zhu, Ermo Hua, Yihao Liu, Zonglin Li, Huayu Chen, Xiaoye Qu, Yafu Li, Weize Chen, Zhenzhao Yuan, Junqi Gao, Dong Li, Zhiyuan Ma, Ganqu Cui, Zhiyuan Liu, Biqing Qi, Ning Ding, and Bowen Zhou. A survey of reinforcement learning for large reasoning models, 2025. URL https://arxiv.org/abs/2509.08827.

[18] Terry Yue Zhuo, Dingmin Wang, Hantian Ding, Varun Kumar, and Zijian Wang. Training language model agents to find vulnerabilities with ctf-dojo, 2025. URL https://arxiv.org/abs/2508.18370.

[19] Terry Yue Zhuo, Dingmin Wang, Hantian Ding, Varun Kumar, and Zijian Wang. Cyber-zero: Training cybersecurity agents without runtime, 2025. URL https://arxiv.org/abs/2508.00910.