Hierarchical Multi-agent Reinforcement Learning for Cyber Network Defense

Extended Abstract

Aditya Vikram Singh
Northeastern University
Boston, United States
singh.adityav@northeastern.edu

Ethan Rathbun
Northeastern University
Boston, United States
rathbun.e@northeastern.edu

Emma Graham
Dartmouth College
Hanover, United States
emma.graham.th@dartmouth.edu

Lisa Oakley
Northeastern University
Boston, United States
oakley.l@northeastern.edu

Simona Boboila
Northeastern University
Boston, United States
m.boboila@northeastern.edu

Peter Chin
Dartmouth College
Hanover, United States
peter.chin@dartmouth.edu

Alina Oprea
Northeastern University
Boston, United States
a.oprea@northeastern.edu

ABSTRACT

Multi-agent Reinforcement Learning (MARL) offers new opportunities in the cyber defense domain. We propose a hierarchical MARL architecture that decomposes defense strategies into specialized sub-tasks like network investigation and host recovery. A master defense policy coordinates these sub-tasks, enabling efficient adaptation to shifting attacker strategies with minimal fine-tuning. Evaluation in the CybORG CAGE 4 cyber defense environment shows that our hierarchical learning approach achieves high performance in terms of convergence speed, episodic return, and several interpretable metrics relevant to cybersecurity.

KEYWORDS

Multi-agent reinforcement learning; Cybersecurity; Deep reinforcement learning; Hierarchical reinforcement learning

ACM Reference Format:
Aditya Vikram Singh, Ethan Rathbun, Emma Graham, Lisa Oakley, Simona Boboila, Peter Chin, and Alina Oprea. 2025. Hierarchical Multi-agent Reinforcement Learning for Cyber Network Defense: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

Cyber defense is critical in both private and public network infrastructures, which are frequently targeted by increasingly sophisticated external attackers with malicious intentions. Although a range of machine learning (ML) tools are available to detect specific classes of attacks [1–3, 7, 8, 11], the advancement of deep reinforcement learning (DRL) presents an opportunity to automate the cyber defense strategy and reduce the burden on security operators.

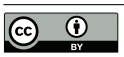
The most recent CAGE 4 challenge [4], by The Technical Cooperation Program (TTCP), offers a realistic environment for studying multi-agent defense strategies. The CAGE 4 challenge leverages the Cyber Operations Research Gym and models a team of multiple blue agents defending a distributed network, playing against multiple red agents compromising the network. Existing techniques for single defensive agents [6, 10] are either computationally expensive or do not generalize to new attackers, and thus cannot be immediately applied to the multi-agent CAGE 4 environment.

In this paper, we propose a scalable multi-agent reinforcement learning (MARL) technique for automating defense in cybersecurity environments such as CAGE 4. Our method decomposes the complex cyber defense task into smaller sub-tasks, trains sub-policies for each sub-task using PPO enhanced with domain expertise, and finally trains a master policy that coordinates the selection of the sub-policies at each time step. We enhance the agents' observation space to store alerts persistently and include security indicators that enable faster response to an ongoing attack. We evaluate our hierarchical techniques against multiple baselines and adversarial behaviors and propose new interpretable metrics that show significant improvements over traditional MARL approaches.

2 PROBLEM STATEMENT

Cyber networks are often segmented into operational enterprise networks that encompass multiple security zones depending on proximity to critical resources. This setup leads to a multi-agent competitive environment, where each defender agent is protecting its own security zone(s), with the overarching team goal of defending the entire network.

The two teams are represented by multi-agent systems: defender (the blue team) and attacker (the red team). The attacker's goal is to maximize its reward by degrading services available to users,



Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

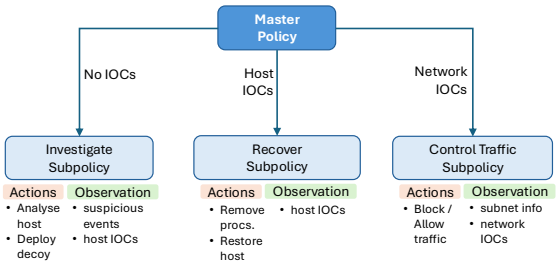


Figure 1: Instantiating H-MARL in cybersecurity. The master policy uses state abstractions related to the presence of IOCs.

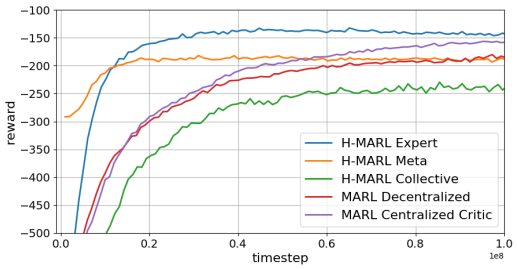


Figure 2: Average training return for all algorithms.

represented by green agents, and compromising the critical Operational Technology (OT) service. The defender's goals are two-fold: maintain the security of the cyber network by reducing the adversarial presence, and minimize the operational impact on users. The blue team monitors and responds to threats through actions such as analyzing hosts for malware, deploying decoy services, blocking or allowing traffic (at the expense of disrupting the work of green agents), removing malicious processes, and restoring hosts to a secure state. The reward scheme models a zero-sum game, where blue agents incur penalties when green agents are affected due to degraded services becoming inaccessible. In addition, blue agents are penalized when red agents impact the critical OT security service.

Challenges. The environment provides partial observability of red presence and blue agents need to run monitor and analyse actions to discover compromised hosts. The policy space is large, including a set of actions for each host on the network, and the observation space is memoryless. In addition, actions have variable duration, and all blue agents share a common reward, even though each of them protects a different part of the network.

3 H-MARL METHODOLOGY

In our proposed design, each blue agent is represented by a hierarchical architecture consisting of a master policy and several sub-policies as follows. The action space $\mathcal{A}$ is first partitioned into smaller subsets, or classes, chosen using domain expertise. For example, the "recover" class refers to all primitive actions to remove processes and restore machines on the network. Thus, each sub-policy handles one class of primitive actions that will be executed in the network. The master policy must learn the best policy $\pi_m : \mathcal{H} \rightarrow \mathcal{A}_m$ over meta-actions, while each sub-policy $\psi_c : \mathcal{H} \rightarrow A_c$ must learn the best policy over all actions in their respective class of meta-action. Figure 1 illustrates the sub-task partition in a cyber environment such as CybORG CAGE 4. We identify three types of sub-tasks: investigate host, recover host, and control traffic between security zones. Expert knowledge refines sub-policy observations using transformation functions $f_c : O \rightarrow O_c$, whose role is to filter information that is irrelevant to sub-policy c. For example, the sub-policy responsible for restoring machines only needs to know about the hosts that present clear signs of compromise, rather than about all the alerts in the system.

The performance of the master policy depends on the performance of each sub-policy. To account for this, we utilize a two-phase hierarchical training approach. We first define an expert policy called **H-MARL Expert**, which uses deterministic rules generated from domain knowledge to select meta-actions and train the sub-policies to near-optimal performance. In the second phase represented by our **H-MARL Meta** algorithm, the previously trained sub-policies are kept frozen and used to generate primitive actions to train the master policy.

The basic observation vector of CybORG blue agents consists of mission index, subnet information, suspicious processes, and suspicious connections. We expand the observation with Indicators of Compromise (IOC) information per host, which include evidence of a cyber threat. We use two types of IOCs: whether malicious files have been detected on the victim machine and the IP address of the compromised host that issues service requests to a decoy service. We also enhance the observation state with memory to persistently store alerts.

4 EXPERIMENTAL EVALUATION

MARL Baselines. We compare our proposed methods H-MARL Expert and H-MARL Meta with three additional baselines: (1) *MARL Decentralized*, a single policy actor-critic architecture (IPPO) with separate value functions for each agent based on local observations; (2) *MARL Centralized Critic* (MADDPG [5]), a single policy method that uses the global state instead of incomplete agent observations to calculate the joint value function during training; (3) *H-MARL Collective*, a hierarchical architecture that attempts to learn both the master and sub-policies from scratch, simultaneously. In the hierarchical variants, both the master and the sub-policies use IPPO.

H-MARL Performance. In Figure 2, we show that H-MARL Expert achieves the best performance (−129.53 reward) by executing recovery actions promptly before attacks escalate. H-MARL Meta performs similarly to MARL Decentralized (−181.62) but trains a master policy 3-5 times faster by selecting sub-policies rather than learning primitive sub-tasks like recovery or investigation. This hierarchical approach is crucial in scenarios where defining expert rules is challenging. Our hierarchical methods generalize well against different adversaries in the environment, as the Recover policy can be reused without retraining, while the Investigate policy requires only minor fine-tuning. For complete details on algorithms and results please refer to the extended version of the paper [9].

ACKNOWLEDGMENTS

This research was funded by the Defense Advanced Research Projects Agency (DARPA), under contract W912CG23C0031.

REFERENCES

[1] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, II, and David Dagon. 2011. Detecting Malware Domains at the Upper DNS Hierarchy. In *Proceedings of the 20th USENIX Conf. on Security* (San Francisco, CA). USENIX Association, 27–27.

[2] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. 2012. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In *Presented as part of the 21st USENIX Security Symp. (USENIX Security 12)*. USENIX Association, 491–506.

[3] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2012. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[4] TTCP CAGE Working Group. 2023. TTCP CAGE Challenge 4. https://github.com/cage-challenge/cage-challenge-4.

[5] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6382–6393.

[6] Garrett Mcdonald, Li Li, and Ranwa Al Mallah. 2024. Finding the Optimal Security Policies for Autonomous Cyber Operations With Competitive Reinforcement Learning. *IEEE Access* 12 (2024), 120292–120305. https://doi.org/10.1109/ACCESS.2024.3446310

[7] Terry Nelms, Roberto Perdisci, and Mustaque Ahamad. 2013. ExecScent: Mining for New C&C Domains in Live Networks with Adaptive Control Protocol Templates. In *Proceedings of the 22nd USENIX Conf. on Security*. USENIX Association, USA, 589–604.

[8] Talha Ongun, Oliver Spohngellert, Benjamin A. Miller, Simona Boboila, Alina Oprea, Tina Eliassi-Rad, Jason Hiser, Alastair Nottingham, Jack W. Davidson, and Malathi Veeraraghavan. 2021. PORTFILER: Port-Level Network Profiling for Self-Propagating Malware Detection. In *IEEE Conference on Communications and Network Security, CNS 2021, Tempe, AZ, USA, October 4-6, 2021*. IEEE, 182–190. https://doi.org/10.1109/CNS53000.2021.9705045

[9] Aditya Vikram Singh, Ethan Rathbun, Emma Graham, Lisa Oakley, Simona Boboila, Alina Oprea, and Peter Chin. 2024. Hierarchical Multi-agent Reinforcement Learning for Cyber Network Defense. https://arxiv.org/abs/2410.17351

[10] Sanyam Vyas, John Hannay, Andrew Bolton, and Professor Pete Burnap. 2023. Automated cyber defence: A review. *arXiv preprint arXiv:2303.04926* (2023).

[11] Ting-Fang Yen, Alina Oprea, Kaan Onarlioglu, Todd Leetham, William Robertson, Ari Juels, and Engin Kirda. 2013. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In *ACSAC*. 199–208.