Realistic and Lightweight Cyber Agent Training Environment using Network Emulation in Mininet

Chih-Ting Yeh, Himanshu Neema, Daniel Balasubramanian
Vanderbilt University, Nashville, TN, USA

***Abstract*—In this work, we enhance the CybORG [1] framework to replace its abstract network simulation with lightweight emulation using Mininet [2]. CybORG's abstract simulation provides efficiency for training cyber agents, but its low-fidelity lacks realism needed for training real-world cyber agents. On the other hand, full cloud-based network emulation or live subnets both provide high-fidelity but require additional resources. Our approach is to strike a balance between the two by using the lightweight emulation capabilities of Mininet, thereby simultaneously emphasizing efficiency, cost-effectiveness, and realism. The proposed demonstration highlights the advantages of Mininet-based emulation and its use for cyber-agent training and rapid validation with minimal resource requirements. In addition, our environment enables the collection of network utilization metrics that can further improve cyber-agent training. This work not only broadens the applicability of CybORG for cybersecurity research, but also provides a more realistic, scalable, and cost-efficient environment for cyber agent training.**

***Index Terms*—Autonomous Cyber Defense, Cybersecurity, Network Emulation, Reinforcement Learning**

I. Introduction

The evolution of cybersecurity threats demands equally dynamic and sophisticated countermeasures. Training and validating cybersecurity agents in environments that accurately mimic real-world network complexities is crucial for developing effective defense mechanisms. CybORG is a widely used versatile platform for cybersecurity research and agent development. However, its highly abstract network simulation lacks the fidelity necessary for training realistic cyber agents. Further, the integration with cloud-based high-fidelity network emulation using virtual machines introduces prohibitive costs and scalability limitations, posing significant barriers to widespread adoption and innovation. In this demonstration, we present an innovative solution that leverages Mininet, a highly efficient network emulation tool with low resource requirements. By integrating Mininet with CybORG, we provide an alternative emulation framework that not only reduces operational costs but also maintains the realism and complexity necessary for effective cybersecurity training. We demonstrate this solution with an application that highlights the potential of Mininet emulation as a robust, scalable, and cost-efficient platform for validating cybersecurity agents within the CybORG environment. This approach broadens CybORG's applicability and opens new avenues for research in cybersecurity training and simulation.

II. Related Work

CybORG [1] is a prominent platform for cyber defense training, widely recognized for its integrated abstract network simulation and emulation interface. This dual-component approach is designed to bridge the "reality gap" — a generalization challenge that arises when reinforcement learning (RL) agents are trained within simulated contexts. To ensure a seamless transition and consistency in actions, observations, and state transitions across both simulated and emulated environments, CybORG shares a common API between emulation and simulation.

CyGiL [3] offers an interactive, stateless emulation architecture, integrating the MITRE ATT&CK framework to provide high-fidelity environments tailored for RL training in advanced cyber operations.

CyberBattleSim [4] is a research platform for studying automated agents within a simulated, abstract enterprise network. Offering a high-level abstraction of network and cybersecurity concepts, its Python-based Open AI Gym interface facilitates agent training via reinforcement learning algorithms.

Mininet [2] enables lightweight emulation of a full virtual network on a single machine, simulating devices, links, routing, and various network conditions. Mininet's strength lies in its ability to provide a realistic testing environment for network configurations, protocols, and applications without the need for physical hardware. This makes it an invaluable tool for rapid prototyping and testing.

III. Integration Architecture

To extend the common API in CybORG with Mininet and create a more interactive, real-world simulation environment, we developed a modular approach as detailed below:

1) *Mininet Topology Generator*: The Topology Generator module enables creation of network topology in Mininet mirroring the topology used in CybORG simulation. This involves translating the abstract representation of networks, hosts, and connections in CybORG into a concrete implementation that can be executed and manipulated within Mininet.
2) *Mininet Command Line Interface Manager*: This module serves as an intermediary between CybORG actions (from blue and red agents) and Mininet, translating actions into equivalent commands that can be executed within the Mininet CLI. It ensures that actions taken in the simulation have direct, observable effects on the emulated network.

979-8-3503-7594-7/24/$31.00 ©2024 IEEE
DOI 10.1109/DESTION62938.2024.00011

TABLE I: Mapping Abstract Red Actions to Shell Commands.

Abstract Method	Shell Commands
Discover Remote Systems	nmap -sn ¡CIDR_address¿
Discover Network Services	nmap -sV ¡IP_address¿
Exploit Remote Services	ssh -p 21/22 ¡IP_address¿
Privilege Escalation	sudo -u root ¡exploit_command¿

3) *Results Bundling*: After executing actions in Mininet, this module is responsible for capturing the outcomes and effects, translating these back into a format that can be understood and processed by CybORG. This enables a closed feedback loop where actions in the simulation have tangible results in the emulation, and the outcomes of those actions inform future steps in the simulation.

IV. Case Study

Our scenario file adapts a subset of Scenario2 from the CAGE Challenge 2 [5], with a reduced action space to streamline the experiment and ensure that a concise set of actions are available to agents both in CybORG and the Mininet based emulated environment. The mapping of the highly abstract actions in CybORG to corresponding Linux shell commands in the emulation is given in Table I and further described below. In addition, to compare the agent performance in abstract network simulation and emulated environments, we trained a PPO agent using the reduced action space.

- *Discover Remote Systems*:This action utilizes nmap for network scanning, aiming to identify active hosts within a specified subnet. The abstracted output is structured in the following format.
  {"success":"True/False/Unknown", "10.0.10.0/24": {'10.0.10.12', '10.0.10.13', '10.0.10.14', '10.0.10.15', '10.0.10.16'}}
- *Discover Network Services*: This scans uses 'nmap' to scan for open ports and available services on targeted IP addresses. The output reflects realistic scanning results, providing valuable information for further actions.
  { "success": "True", "10.0.214.187": {'21', '22'}}
- *Exploit Remote Services*: The action is named FTPDirTraversal but is implemented an SSH brute force exploit It demonstrates a practical approach to service exploitation, with detailed output including success status, exploited services, and additional technical details.
- *Privilege Escalation*: Executed via the sudo command, this action provides higher privileges on a compromised host. The detailed output further enhances the scenario's realism, offering insights into the success of such exploits and their implications in the emulated environment.

The results presented in Figure 1 exemplify the high degree of correlation between the emulation observations in Mininet and the expected outcomes in the CybORG simulation. This alignment is crucial, as it demonstrates the integrated system's ability to replicate the complex interactions and behaviors observed in cyber-physical environments accurately. The accurate mapping of actions, such as 'DiscoverRemoteSystems', to tangible results — the enumeration of active hosts within a network — not only validates the emulation's precision but also its effectiveness in simulating real-world cybersecurity scenarios.

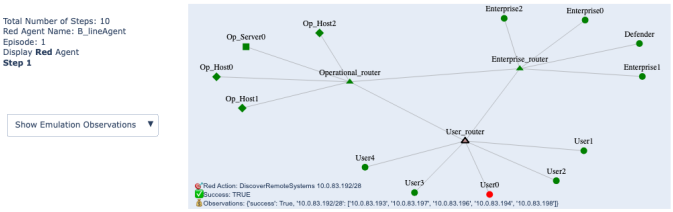

Fig. 1: Visualizing Cyber Game over Emulation in Mininet

V. Conclusion & Future Work

In this paper, we presented a novel approach to integrating the CybORG simulation platform with Mininet, a network emulator, to create a more cost efficient and realistic cybersecurity training and research environment. The modular design comprising the Topology Generator, Mininet CLI Manager, and Results Bundling modules facilitates a seamless transition between CybORG's simulated actions and their real-world counterparts within the Mininet environment. In the future, we plan to extend the repertoire of actions that can be translated from CybORG to Mininet, covering more sophisticated and nuanced cybersecurity maneuvers. We also intend to broaden our framework to include operational technology (OT) scenarios, integrating tools like *GridLab-D* to address the merging realms of IT and OT. This expansion is designed to make our platform more versatile in addressing diverse cybersecurity challenges. Furthermore, a quantitative comparison of our Mininet implementation with alternatives will help pinpoint both its advantages and areas for improvement, guiding our ongoing refinement of the cybersecurity training environment.

VI. Acknowledgements

This work is supported by DARPA through contract number W912CG-23-C-0028. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsor.

References

[1] Callum Baillie, Maxwell Standen, Jonathon Schwartz, Michael Docking, David Bowman, and Junae Kim. Cyborg: An autonomous cyber operations research gym. *arXiv preprint arXiv:2002.10667*, 2020.

[2] Faris Keti and Shavan Askar. Emulation of software defined networks using mininet in different simulation environments. In *2015 6th International Conference on Intelligent Systems, Modelling and Simulation*, pages 205–210. IEEE, 2015.

[3] Li Li, Raed Fayad, and Adrian Taylor. Cygil: A cyber gym for training autonomous agents over emulated network systems. *arXiv preprint arXiv:2109.03331*, 2021.

[4] Microsoft Defender Research Team. Cyberbattlesim. https://github.com/microsoft/cyberbattlesim, 2021.

[5] Cyber autonomy gym for experimentation challenge 2. https://github.com/cage-challenge/cage-challenge-2, 2022. Created by Maxwell Standen, David Bowman, Son Hoang, Toby Richer, Martin Lucas, Richard Van Tassel, Phillip Vu, Mitchell Kiely.