Realistic Pentesting Training Framework for Reinforcement Learning Agents

2210406 NGUYEN Huynh Phuong Thanh

Penetration testing, or pentesting, refers to assessing and enhancing network system security by trying to identify and exploit any existing vulnerabilities. This is one of the critical methods widely used by organizations to strengthen their defenses against malicious attacks. The pentester executed an authorized attack on the network systems to gain administrator permissions, allowing them to evaluate overall security characteristics. However, traditional manual pentesting raises several challenges and becomes ineffective due to its time-consuming nature and the need for technical security skills. With modern network systems becoming more complex and threats increasingly sophisticated, manual pentesting can not adapt slowly.

Applying Artificial Intelligence (AI) techniques to the cybersecurity domain is proposed to solve this problem and enable the automation of pentesting procedures. Reinforcement Learning (RL) was created as an AI optimization technique to produce an attack policy that learns the best strategy through environmental interaction. Due to this learn-via-interaction mechanism, it has recently become an effective method for creating autonomous pentesting agents. These agents are trained to replicate the actions of humans with enhanced speed, scale, precision, and automation. Many proposed approaches involve using simulation environments to train pentesting RL agents. The main advantages of these studies are their speed, low resource consumption, and ease of design, making them a potential solution to shift from manual to automated procedures.

However, due to the logical modeling of attack actions and observations, there is a heavy reliance on predefined constants and probabilistic values for agent actions and environment states. This can lead to inaccuracies in replicating real-world behavior due to unexpected factors, decreasing agent accuracy and performance. Additionally, the simulated network may not accurately represent the configuration and topology of an actual network. Thus, simulation environments for training RL pentesting agents present challenges when deploying them in actual network infrastructure due to the lack of realism in the simulation-trained agents.

We propose PenGym, a framework for training pentesting RL agents in realistic environments, to address this issue. The most significant features of PenGym are its support for real pentesting actions, full automation of the network environment creation, and good execution performance. PenGym covers network discovery and host-based exploitation actions that are avail-

able to train, test, and validate RL agents in an emulated network environment. Compared to typical simulation-based agent training, the main advantage is that PenGym can execute actual actions in a real network environment while providing a reasonable training time. We conducted several experiments to demonstrate the effectiveness of using PenGym as a realistic training environment compared to a simulation approach (NASim).

For the smallest scenario, agents trained in both simulation and emulation environments achieved equivalent results with minor differences. For the mid-size scenario, simulation-related limitations occurred. Although the agents trained in the NASim environment performed well in the simulation environment, they were ineffective when tested in the emulation environment, showing high variation and large attack step counts. In contrast, after fixing logical modeling issues in the simulation to create the revised version NASim(rev.), testing results were comparable to PenGym in both the simulation and emulation environments.

For the largest scenario, the effectiveness of the PenGym approach is emphasized. Agents trained in the original NASim environment behaved poorly when tested in a real environment, having a high failure rate. In contrast, agents trained in PenGym successfully reached the pentesting goal in all our trials. Even though NASim(rev.) was revised with a more accurate model, experiment results with the largest scenario indicated that agents trained in PenGym slightly outperformed and were more stable than those trained in NASim(rev.). Thus, the average number of step differences required to reach the pentesting goal ranged from 1.4 to 8, which is better for PenGym. Consequently, PenGym provides a reliable and realistic training environment for pentesting RL agents, eliminating the need to model agent actions via simulation. This finding justifies the use of realistic environments for creating and training RL agents for pentesting purposes.

Regarding time performance, due to the actual action execution on the cyber range, PenGym requires more training time than simulation environments. However, it provides a reasonable training time in more complex scenarios while preserving realism and feasibility as real networks become more sophisticated. In particular, for the largest scenario and most intricate RL algorithm, PenGym training takes around $17,000$ s compared to $14,000$ s in NASim, with a ratio of roughly 1.2.

Future work involves proposing a realistic automatic scenario generator to assist in constructing a realistic pentesting scenario for training RL agents.

Keywords: Penetration Testing, Reinforcement Learning, Agent Training Environment, Realistic Environment, Cyber Range, Cybersecurity