INTRODUCTION

Two teams are needed to work on transforming a virtualized network into an open AI gym interface for cyber-attack/defense scenarios. The virtual network was built collaboratively by both teams. The network was completely specified in software to allow for easy resetting of the environment at the conclusion of each episode, and user simulation agents will be used to create background noise to mask adversarial impacts.

Responsibilities were divided into red and blue teams after the infrastructure was completed. The red team would be in charge of creating an interface between an OpenAI gym and a G2 platform for attackers like Metasploit or Covenant. This interface covers lateral movements into another system, data exfiltration, and ransomware payload execution. The blue team’s job was to link a defensive agent to the environment. This agent should be able to obtain current host/network alerts, do investigative tasks with a tool like VelociRaptor or osquery, and perform remedial tasks like terminating processes or reimaging computers.

AIM

Companies employ penetration testing to check the security of their software and infrastructure. Testing involves a group of security experts acting as attackers to find security flaws before hackers do. The purpose of a pen tester is to inform a corporation about its vulnerabilities. In the world of security this is commonly referred to as red teaming. On the other hand, the company’s security team, the blue team, uses penetration testing to determine which areas of security need to be strengthened.

Our CISC-499 project focused on working as the red team for the LINNA laboratory. We implemented machine learning algorithms to explore the vulnerability of the blue team’s system. We also worked with CybORG, a work-in-progress gym for ACO (Autonomous Cyber Operations: development of decision-making agents in adversarial scenarios) research, to assist the deployment of machine learning methods to solve this challenge and to encourage researchers in this field to pay attention to problems in the ACO environment. CybORG is a simulation and emulation environment with a common interface that allows for quick training of autonomous agents before they are tested on real-world systems. The feasibility of this method has been demonstrated in preliminary testing. Essentially, we implemented another emulation backend using another library known as Cyber range to work under the CybOrg interface.

METHODS

We first studied the CybORG codebase and interface. This involved reading the project’s Github and going through the tutorial notebooks. We also studied the Cyber Range interface and configurations for emulated network/host creation. This involved creating test environments and running experiments. We also implemented automated emulation environment following the CybORG API using the Cyber Range interface. This would involve coming up with descriptions for the CybORG agents and other environment variables and system images. Environment reset/updates are enabled in these scenarios to prepare for training purposes. Implement a set of attack actions under the emulation interface. This would involve coming up with commands and parameters that the CybORG agents can follow. We would also test the performance of the agents in this process.

CONCLUSION

Overall, this project has taught us that there is a massive potential in the use of machine learning in cybersecurity. The learning properties of algorithms allows for security systems to incrementally become better at protecting themselves from exploits.

We also practiced using version control with Git. We used Github issues to track milestones. We also deeply understood the importance of proper documentation.