Reinforcement Learning for Cyber Red Teaming

Overview

- Cyber red teaming is the simulation of an attack on one's own computer network to identify security weaknesses
- Red teaming is time-consuming—can reinforcement learning (RL) help?

Challenges with RL for Red Teaming:

- Most network data is naturally represented as a graph that changes as it's explored
- High-level strategic decision making is required in response to human red-teamer instructions

Background

Reinforcement learning (RL) trains agents to autonomously play complex "games." Using RL requires specifying three important ingredients:

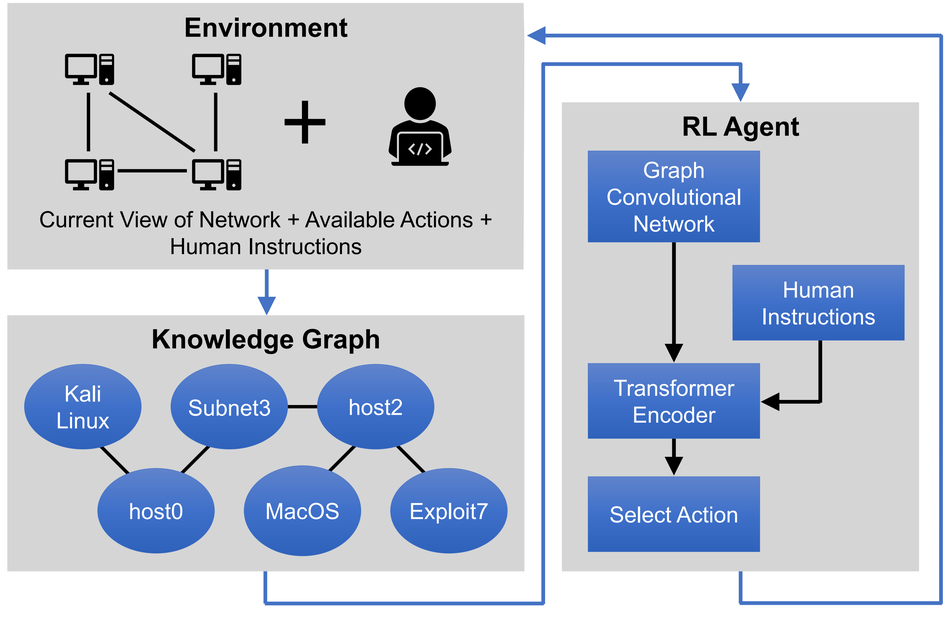
Environment
How does the agent perceive/ interact with the environment?

Architecture
How does the agent "think" about the environment?

Training
What does the agent experience during training?

Approach

Challenge	Our Solution
RL agent assists human red-teamer, so network data should be structured to be understandable to agent and human	Current view of network is processed into a knowledge graph representing connections between entities on network as well as their attributes
Agent needs to process relationships between connected nodes in knowledge graph	Use a graph neural network
Nodes are frequently added to the knowledge graph	Use a graph convolutional network (GCN) architecture that can process unseen nodes
Agent needs to process data that cannot be represented in the knowledge graph, such as human red-teamer instructions	Use a transformer encoder after the GCN so agent can attend to relevant human instructions

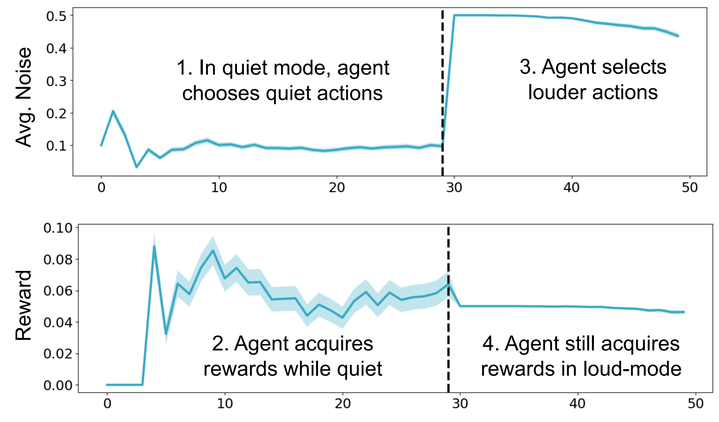


Results

Simulated Red-Team Scenario:

Actions generate "noise." Human red-teamer can prompt agent to go quiet or loud

- Go quiet: agent kicked off network after too much noise
- Go loud: agent gets reward for being loud



Impact

We trained RL agents to provide recommended actions during cyber red teaming. The agent architecture is designed to handle the challenges of operating on a computer network

Future Work:

- Improve realism of training simulator
- Understand limits of agent architecture

References: Hamilton, Ying, Leskovec. "Inductive Representation Learning on Large Graphs." *NeurIPS*. 2017; Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin. "Attention is all you need." *NeurIPS*. 2017

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Jensen Dempsey, Whitney Pavlova, Tim Reid, and Will Stephenson
Group 52 - AI Technology & Systems, MIT Lincoln Laboratory, Lexington, MA