The Center for Cyber Defenders

Expanding computer security knowledge

Adversary Emulation with Planning AI

Nolan Bonnie - University of California, Irvine

Project Mentor: Vince Urias, Org. 9315

Introduction to Emulation

Cybersecurity researchers want to apply machine learning to automate adversary detection. However, data for training these models is scarce and often times unrealistic. There is a need for representative APT data.

The Problem:

- Not enough data to train models
- Available data is unrealistic

The Solution:

- Use AI planning to construct attack paths
- Execute the attack actions on virtual machines

Objectives and Approach

In order to properly emulate an adversary, we need a system that can plan out actions to achieve its goal. Such a system is broken into 3 main components:

- An agent (the adversary)
- Actions that the agent can perform
- An environment the agent can interact with

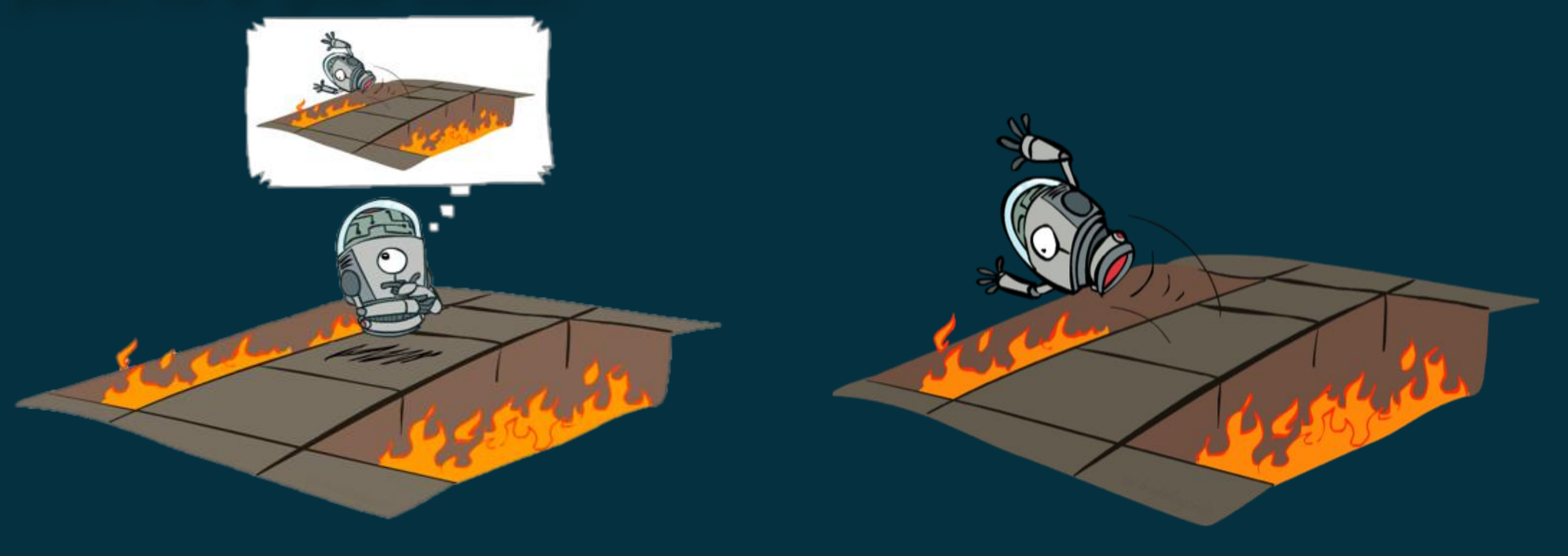
We define our desired network topology in our environment, and our agent plans a path between its initial state and the goal state using only the actions provided.

Finite State Machines vs Planning

A common way to get an agent to a goal is by implementing a finite state machine (FSM). A FSM is able to string together logical sequences to achieve a goal, but lacks flexibility and reality, as action chains are pre-defined. Planning algorithms are less structured, simply give an agent a list of actions, and the agent will figure out how and when to use them to achieve its goal.

AI Planning

AI Planning splits the difference between a Monte Carlo simulation and reinforcement learning. Specifically, our planner implements a variant of the A* informed search algorithm, which uses heuristics to traverse the state-action tree. A* will aggressively (and accurately) prune a full Monte Carlo state-action tree until it finds the optimal path to a goal state.

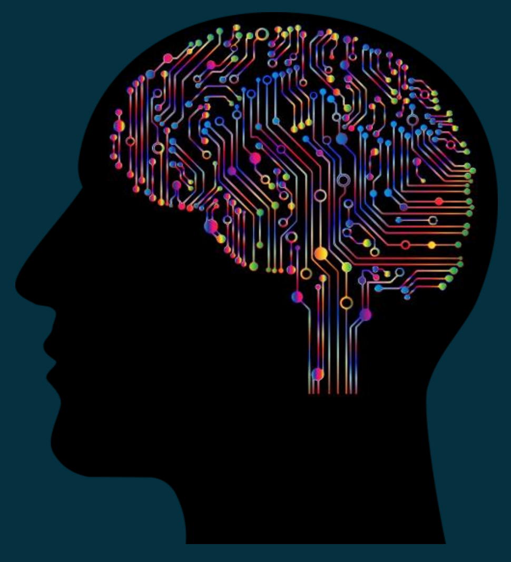


Motivation

The planner algorithm is based off Goal-Oriented Action Planning (GOAP), a powerful planning AI that has mainly been implemented in strategic computer games to guide NPC decisions. With some modifications, we can take the same concepts and apply them to emulating an adversary.

Purpose

The GOAP planner has gone through several iterations to meet the requirements of adversary emulation. This project will be one of the first implementations of GOAP in Python, as well as the first occurrence of applying GOAP to adversary emulation.



Goal

The goal of this project is to produce realistic system data of different APT groups. When we know what an adversary in our system would look like, we can better protect ourselves from future attacks. Additionally, our project aims to generate large quantities of realistic data, which will hopefully open the door to future cyber machine learning applications.

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Future Work

The next step for this project is expressing APT groups as agents in our environment. This will allow our AI to strategically plan an adversary's attack on a given network topology.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2020-7807 D.